



(19) Országkód

**HU**



**MAGYAR  
KÖZTÁRSASÁG**

**MAGYAR  
SZABADALMI  
HIVATAL**

# **SZABADALMI LEÍRÁS**

(11) Lajstromszám:

**219 533 B**

(51) Int. Cl.<sup>7</sup>

**G 06 F 13/28**

(21) A bejelentés ügyszám: P 94 00792  
(22) A bejelentés napja: 1992. 08. 26.  
(30) Elsőbbségi adatok:  
07/761,534 1991. 09. 18. US  
(86) Nemzetközi bejelentési szám: PCT/EP 92/01965  
(87) Nemzetközi közzétételi szám: WO 93/06553

(40) A közzététel napja: 1995. 05. 29.  
(45) A megadás meghirdetésének dátuma a Szabadalmi  
Közlönyben: 2001. 05. 28.

(72) Feltalálók:

Carmon, Donald Edward, Durham,  
North Carolina (US)  
Crouse, William George, Raleigh,  
North Carolina (US)  
Ware, Malcolm Scott, Raleigh,  
North Carolina (US)

(73) Szabadalmaz:

International Business Machines Corp., Armonk,  
New York (US)

(74) Képvisező:

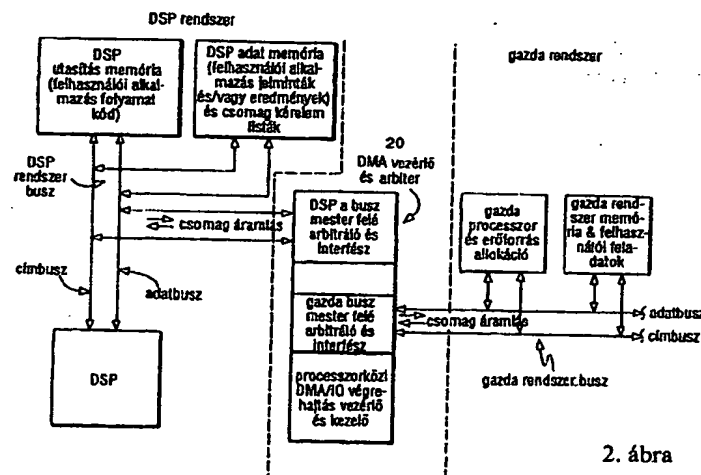
dr. Bogsch Attila, Budapest

## **(54) Multimédia-számítógéprendszer, valamint eljárás multimédia számítógéprendszer működésének vezérlésére**

### **KIVONAT**

Multimédia-számítógéprendszer, amely felhasználóialkalmazás-programok végrehajtásához egy, azok végrehajtásának támogatása során a rendszerhez digitális jelfeldolgozási kérélmeket továbbító első processzort, továbbá a felhasználóialkalmazás-programok első processzor általi végrehajtásának támogatása során a digitá-

lis jelfeldolgozási kérélmeket végrehajtó második processzort (DSP), továbbá a második processzor (DSP) által fogadott digitális jelfeldolgozási kérélmek függvényében a második processzort (DSP) a közvetlen memória-hozzáférésű adatátviteli kérélmek particionált vá-  
rakozósorának a digitális jelfeldolgozási kérélmek tá-



2. ábra

A leírás terjedelme 30 oldal (ezen belül 13 lap ábra)

**HU 219 533 B**

mogatása során az adatok második processzor (DSP) felé vagy processzortól való továbbítása céljából történő létrehozására utasító eszközt tartalmaz, amelynek egy, az első processzortól a második processzor rendelkezésre álló jelfeldolgozásierőforrás-igényükkel a második processzorhoz (DSP) továbbítására utasító eszköze van, valamint a particionált várakozósor a második processzor (DSP) által szabályosan visszatérő időközönként behelyezett particiójelöléseket tartalmaz.

Eljárás felhasználóalkalmazás-programokat végrehajtó első processzort és a felhasználó alkalmazási programoknak az első processzorban való végrehajtásának támogatása során digitális jelfeldolgozási kérélmeket végrehajtó második processzort tartalmazó multimédia-számítógéprendszer működésének vezérlésé-

re, amelynek során felhasználó által kiválasztott aktív alkalmazási programok végrehajtásának támogatására meghatározzák a második processzor (DSP) teljes erőforrásigényét és összehasonlítják a második processzor (DSP) legnagyobb rendelkezésre álló erőforrás-kapacitásával, és ha a felhasználóalkalmazás-programok erőforrásigénye következtében a második processzor (DSP) teljes erőforrásigénye meghaladja a második processzor (DSP) legnagyobb rendelkezésre álló erőforrás-kapacitását, akkor megtöltik a kiválasztott felhasználóalkalmazás-program első processzor általi végrehajtását, továbbá a második processzorral (DSP) vett digitális jelfeldolgozási kérélmek függvényében közvetlen memória-hozzáférésű adatátviteli kérelem particionált várakozósort hoznak létre, amelyben szabályosan ismétlődő időközönként particiójelöléseket helyeznek el.

A találmány tárgya egyrészt egy multimédia-számítógéprendszer, amely felhasználóalkalmazás-programok végrehajtásához egy, a felhasználóalkalmazás-programok végrehajtásának támogatása során a rendszerhez digitális jelfeldolgozási kérélmeket továbbító első processzort, továbbá a felhasználóalkalmazás-programok első processzor általi végrehajtásának támogatása során a digitális jelfeldolgozási kérélmeket végrehajtó második processzort, továbbá az utóbbi által fogadott digitális jelfeldolgozási kérélmek függvényében a második processzortól a közvetlen memória-hozzáférésű adatátviteli kérélmek particionált várakozósorának a digitális jelfeldolgozási kérélmek támogatása során az adatok második processzor felé vagy a felől való továbbítása céljából történő létrehozására utasító eszközt tartalmaz. A találmány tárgya másrészt egy felhasználóalkalmazás-programokat végrehajtó első processzort és a felhasználó alkalmazási programoknak az első processzorban való végrehajtásának támogatása során digitális jelfeldolgozási kérélmeket végrehajtó második processzort tartalmazó multimédia-számítógéprendszer működésének vezérlésére vonatkozó eljárás.

A találmány tárgya általánosabban multimédia-számítógéprendszerben általánosan fellépő dinamikus, szorosan valós idejű, konkurens jelfeldolgozó igények kezelése.

A jelfeldolgozó processzorok számos, napjainkban kapható számítógéprendszer ismert alkotórészei. Számos gyártó gyárt és forgalmaz célfeladatokra orientált speciális digitális jelfeldolgozó processzorokat, közt ismert – és leírásunkban is használt – angol rövidítéssel DSP-ket, amelyeket nagy sebességű, iteratív algoritmusok végrehajtásánál alkalmaznak. Ilyen algoritmusok jelentkeznek például a digitális jelszűrésnél, a beszédfelismerésnél vagy beszédsszintetizációnál, a szervomechanizmusok vezérlésénél, a kódolt beszédgenerálásnál, lézerlemez hifi-mintavételezett hang és zene létrehozásánál, modem adatmodulációs és demodulációs feladatoknál, telefaxadat-átviteli kódolási és -dekódolási

feladatoknál, színes és monokróm képadatok komprimálási és kijelzési feladatainál, mozgókép-feldolgozó feladatoknál, valamint számos adatprotokoll-átalakításnál vagy kódolási, hibajavítási vagy hasonló feladatoknál. A gyakorlatban ilyen algoritmusok, mint például Fourier-transzformációk stb. nagy sebességű, ismételt végrehajtására alkalmas potenciális jelfeldolgozó-processzor-feladatok fenti listáját az iparban széles körben alkalmazzák nagy sebességű, nagy mennyiségű digitális vagy analóg jelmintáknál.

Különösen előnyös jelfeldolgozóprocesszor-architektúrát ismert az US 4 794 517 számú szabadalmi leírás, amelynek tulajdonjoga jelen bejelentésre szállt át, és amelyet annak magyarázatára, hogyan épülnek fel és működnek a digitális jelfeldolgozó processzorok, jelen leírásunkban hivatkozási alapként tekintünk. A területen jártas szakember számára jól ismert, hogy a fent említett jelfeldolgozóprocesszor-architektúra csupán egy a piacon kapható számos ilyen architektúra közül. Ehhez járul még, hogy a digitális jelfeldolgozó processzorok felhasználása mértani függvény szerint növekszik, elsősorban az úgynevezett multimédia-számítógéprendszerek erőteljesen növekvő területén. Ilyen rendszerekben a felhasználó sok esetben egyidejűleg kíván számos különböző feladatot végrehajtani, például átvitel céljára beszéd kódolást, mozgóképeket, modemátvitelt és vételt és talán a háttérben még CD-lemez-lejátszást, hogy csupán néhány audiovizuális vagy multimédiaalkalmazást említsünk. Ezek az alkalmazások egy tipikus gazdarendszeren, például a IBM cég Personal System/2 számítógépcsaldjának egyik tagján, vagy a napjainkban széles körben áruolt, hasonló felépítésű, számos, konkurens adatfeldolgozásra képes számítógépen futtathatók.

Az ilyen multimédiarendszerekben a jelfeldolgozási feladatok általában közvetlen memória-hozzáférés, a leírásunkban is használt közismert angol rövidítéssel DMA útján az erre specializált nagy sebességű digitális jelfeldolgozó processzorhoz kerülnek át. Ahogy a processzorok sebessége és tárolókapacitása növekszik, a di-

gitális jelfeldolgozó processzorok átvehetik maguknak a felhasználói feladatoknak a végrehajtását is, amint azt a későbbiekben kifejtjük. Valójában egy ilyen rendszerben a DMA feladatokat önmagukban egy külön, második processzor is lekezelheti. Ha csupán néhány DMA csatornára van szükség, akkor egy tipikus DMA vezérlő engedélyezheti az első processzornak és a második processzornak, hogy egy vagy több különböző hardvereszközt vezényeljen. Ott azonban, ahol az első processzor többfeladatos feldolgozást végző processzor, és nagyszámú bemeneti/kimeneti (leírásunkban is használt közismert angol rövidítéssel I/O) eszköz van jelen, a többfeladatos feldolgozást végző első processzor és az őt támogató második processzor közötti kisszámú DMA csatorna szűk keresztmetszetté válhat, amely megoldhatatlan problémát jelent olyan esetekben, ha a rendszer DMA kapacitását az összes működő I/O eszköz, valamint az egymás mellett futó különböző alkalmazás egyszerre kívánja igénybe venni.

Képzeljünk el egy olyan jellemző multimédiakörnyezetet, amely számos HIFI audiojelmintát tartalmaz, amelyek feldolgozása a memóriában tárolt független gazda-audiointamemóriátáblák, többszörös memóriavezérlő táblák és amplitúdó- és burkológörbe-táblák segítségével történik, és olyan rendszerre van szükség, amely 1  $\mu$ s-nál kisebb időtartamon belül több mint 100 külön DMA csatornát képes rendelkezésre bocsátani. 16 HIFI sztereoaudiojel-csatorna esetén mindegyik audiocsatorna másodpercenként 88 200 bájtnyi minta átvitelét igényli. Ilyen rövid időn belül is, 100 DMA csatorna több mint 100 000 blokkátvitelt támogat másodpercenként, ahol mindegyik blokknak megvan a saját egyedi forrás- és célcíme és blokktranszfermérete. Hagyományos DMA eszközt használva egy rendszer processzorát, például egy második processzort minden egyes új adatátvitelhez meg kellene szakítani, hogy átlagosan 10  $\mu$ s-onként támogassa a kívánt műveletet. Több gépi ciklusra lenne szükség minden egyes átvitel megvalósításához, és a rendszer hamarosan működésképtelenné válna, mivel nem maradna processzor-erőforrás az aktuális jelfeldolgozó feladatok végrehajtására a DMA hozzáférések szükséges vezérlésén túl.

A jellemzőnek mondható audioalkalmazások mellett számos más olyan, a gazdagép által végrehajtandó alkalmazás létezik, amely kommunikációt igényel a második processzor és az első processzor között, miközben audiojeleket „játszik le”. Erre jó példa lehet egy olyan faxmodem, amely képadatokat visz be a gazdagéprendszer memóriájába, vagy vesz ki onnan, a háttérben egyidejűleg játszott zene, beszéd vagy más audiojelek mellett. Ehhez járulhat továbbá egy olyan beszédfelismerő alkalmazás is, amely a gazdarendszer-memóriába, valamint a memóriából a jelfeldolgozó processzor memóriájába beszédsablonok mozgathatóságát igényli az azonosítási feladat elvégzése céljából, majd vissza kívánja írni a beszédfelismerési tokeneket azok felismerése után. Mindez nem elég; a rendszerben futhat ezenkívül egy beszédsszintetizáló alkalmazás is, amely fonémaadatokat vételez a rendszer memóriából, míg egy CAD alkalmazásban a jelfeldolgozó processzor képtárolóban

lévő háromdimenziós tárgy forgatását végzi, ami közismerten számos komplex iteratív számítást igényel. Mindennek röviden az az eredménye, hogy a jelfeldolgozó processzor által támogatott összetett multiprocesszoros és konkurens adatfeldolgozó környezetben nagyszámú hatékony DMA csatornára van szükség.

Például az US 4 831 523 számú szabadalmi leírásból olyan összetett DMA vezérlőeszközt ismerhetünk meg, amely periférius eszközvezérlőként működik, és rögzített számú – az említett referencia esetében négy – fizikai perifériális eszközt képes a rendszer buszára csatlakoztatni. A fizikai eszközök nem azonosak azokkal a konkurens adatfeldolgozó eljárásokkal, amelyek szoros valós idejű feldolgozott jelmintákat igényelnek a felhasználói kíváncságnak megfelelő feladat végrehajtására. Négy fizikai periférius eszköz hozzávetőlegesen 8 logikai DMA csatorna határát jelenti, és az említett szabadalmi dokumentumban az eszközök nem rögzített időosztásos módon kerülnek megszállításra, mert a DMA vezérlő az eszközöket gyűrűsen, szabadon kezel és bármelyik eszköz meghatározatlan hosszú időt vehet igénybe a feladata elvégzéséhez, elvéve ezzel a lehetőséget a többi valós idejű eszköztől arra, hogy az adatát egy rögzített időtartamon belül befejezhesse. DMA csatornamegosztó mechanizmust ismertet ezen túlmenően az IBM Technical Disclosure Bulletin folyóirat 30. évfolyam 7. száma. Ez az 1987 decemberében megjelent folyóirat 369. és 370. oldalán olyan mechanizmust ismertet, amely dinamikusan újra hozzárendelhető I/O eszközhordvert igényel. Úgy tűnik, hogy a bemutatott mechanizmus olyan szoftver jellegű megoldás, amely kisszámú hardver DMA csatornát dinamikusan hozzárendel olyan külső hardvereszközökhöz, amelyek lehetővé teszik a DMA csatornák megosztását. Nincs meghatározott információ nagyság, szám vagy időtartam tekintetében arról, hogy milyen esetben biztosított az összes kérelem kielégítése, és nem ismertet részleteket, hogy a hardvereszközök milyen módon hozzak létre vagy továbbítják ezt a kérelmet. A bemutatott rendszer természetében nem valós idejű, azaz olyan feladatokat igényel, melyek jelmintáit úgy kell feldolgozni és továbbítani, hogy ennek során ne kötődjön pontos és ismétlődő időnövekményekhez, például CD-zene-reprodukciós rendszer kiszolgálását 88 200 bájtnyi információval másodpercenként, amelyet a kérelmezőfeladatnak át kell vinnie, feldolgoznia majd a feldolgozott jeleket használatra vissza kell továbbítania.

Az US 4 807 121 számú szabadalmi leírás periférius interfészrendszert ismertet, amelynek I/O processzora legfeljebb négy multiplexer egységhez csatlakozik, és minden egyes multiplexer egység legfeljebb négy vezérlőhöz biztosít interfészt. Az I/O processzornak DMA csatornája van, amely a multiplexer egységektől multiplexelt soros adatokat fogad. Az adatok az I/O processzor és bármely vezérlő között úgy kerülnek átvitelre, hogy az I/O processzor helyi memóriájából sorosan, egy DMA csatornán át egy puffertárolóterületét töltik fel. Csupán egyetlen DMA csatorna létezik, és a multiplexelés teszi lehetővé ennek az egy csatornának az alkalmazását. Az adatrészek a vezérlőtől a multiplexer

egységekhez időosztásos alapon jutnak át, és a multiplexer egységek tárolóiból a memóriába sorosan kerülnek át. A dokumentum nem ad azonban kitanítást arra, hogy ez a rendszer bármilyen módon támogatni tudná azt a dinamikus, szoros, valós idejű igényt, amely lehetővé tenné a fentiekben vázolt alkalmazói feladatok egyidejű, szimultán végrehajtását és elvégzését.

A fent leírtak fényében célunk a találmánnyal, hogy olyan hatékony multimédia-számítógéprendszert és adatátviteli mechanizmust hozzunk létre, amellyel lehetővé válna egy első processzorban a szoros, valós idejű konkurens adatfeldolgozási műveletek támogatása.

A fent vázolt problémák megoldására a találmány értelmében a jelfeldolgozó processzor számára feladatkérelmeket állítunk elő az első processzortól, analizáljuk a beérkező feladatkérelmeket, és ebből felépítünk a memóriában egy particionált várakozósorban egy csomagátvitelkérelem-listát. Ezt a particionált várakozósor egy processzor közti DMA vezérlő és arbiterrel megszólítjuk, és a szükséges adatjelmintákat egy rögzített minimális előírt időközön belül bevisszük vagy kivisszük. Az adatátvitelcsomagkérelem-listákat a második processzor hozza létre az adott kiviteli alakhoz szükséges DMA vezérlőcsomagok formájában.

Mindegyik csomagkérelem több vezérlőinformáció-szót tartalmaz, valamint az adatminták forráscímét és mozgathatóságához szükséges célcímét. Az egyik cím az az érintett rendszermemória-cím, amely adat- vagy folyamatminták vételére szolgál, vagy ilyeneket tartalmaz. A második cím az az érintett helyi második processzorinstrukció vagy adatmemória-cím, amelyre vagy amelyből a nyers jelminták, vagy a feldolgozott jelminták beírásra vagy kiolvasásra kerülnek. A csomagokba beépített vezérlőinformáció több adat- vagy utasításbitet tartalmaz, amelyek a mozgatható adatblokk méretét és az áramlás irányát jelzik, azaz azt, hogy az információ a második processzortól vagy a második processzor felé folyik. A második processzor operációs rendszere által létrehozott DMA csomaglistát a második processzor adatmemóriája őrzi, és szekvenciálisan a második processzor buszával és az első processzor buszával interfészkapcsolatban álló DMA hardver szólítja meg és hajtja végre. A DMA processzor a csomagkérelmek particionált listájában lévő egy-egy partició teljes tartalmát megszólítja és végrehajtja szabályos időközönként, amelyek az első processzorban futó bármely lehetséges kiválasztott felhasználói feladat maximális igényeihez igazodnak. Például sztereó CD-lemezjátszókról audiolejátszás céljára digitális zenei minták feldolgozásának támogatására 88 200 16 bites mintát kell a jelfeldolgozó processzorhoz eljuttatni, feldolgozni és a digitális-analóg konverterhez vagy pedig az első processzorhoz eljuttatni egy szabályosan ismétlődő alapon, minden egyes másodpercben.

A kitűzött feladat megoldása során olyan multimédia-számítógéprendszerből indultunk ki, amely felhasználóalkalmazás-programok végrehajtásához egy, a felhasználóalkalmazás-programok végrehajtásának támogatása során a rendszerhez digitális jelfeldolgozási kérelmeket továbbító első processzort, továbbá a

felhasználóalkalmazás-programok első processzor általi végrehajtásának támogatása során a digitális jelfeldolgozási kérelmeket végrehajtó második processzort, továbbá az utóbbi által fogadott digitális jelfeldolgozási kérelmek függvényében a második processzort a DMA adatátviteli kérelmek particionált várakozósorának a digitális jelfeldolgozási kérelmek támogatása során az adatok második processzor felé vagy a felől való továbbítása céljából történő létrehozására utasító eszközt tartalmaz. A továbbfejlesztés értelmében az első processzort a csupán együttes jelfeldolgozásierőforrás-igényükkel a második processzor rendelkezésre álló jelfeldolgozásierőforrás-kapacitását meg nem haladó digitális jelfeldolgozási kérelmek második processzorhoz továbbítására utasító eszköze van, valamint a particionált várakozósor a második processzor által szabályosan visszatérő időközönként behelyezett particiójelöléseket tartalmaz.

A találmány szerinti számítógéprendszer egy további előnyös kiviteli alakja értelmében a DMA adatátviteli kérelmeket a memória forrás- és célcímeket, átviteli irányt, átvicendő adategységek számát, és működési vezérlőjelöléseket, valamint kérelem vége jelölést meghatározó adatokat tartalmazó mezőkkel ellátó második processzort tartalmaz.

Ugyancsak előnyös a találmány értelmében, ha az első processzor és a második processzor közötti adatmozgatást vezérlő DMA vezérlő és arbiter, valamint azt az első processzorral és a második processzorral összekötő adatbuszt és címbuszt tartalmaz.

Előnyös a találmány értelmében továbbá, ha a particionált várakozósor elérése, valamint az adatmozgatásnak a kérelembe épített adatokat tartalmazó mezővel összhangban történő mozgathatóság vezérlése révén az első processzor és a második processzor között adatátvitelt végrehajtó DMA vezérlő és arbiterrel rendelkezik.

A találmány értelmében ugyancsak előnyös, ha egy várakozósor particiót létrehozó, és abba a várakozósor másik particiójából származó adatátviteli kérelmeken alapuló adatátvitelt végrehajtó DMA vezérlő és arbiter adatátviteli kérelmeit szabályos időközönként elhelyező második processzorral rendelkezik.

A találmány szerinti számítógéprendszer egy további előnyös kiviteli alakja értelmében a felhasználó által kiválasztott minden egyes felhasználóalkalmazás-program támogatásához szükséges második processzor jelfeldolgozásierőforrás-igényekhez a felhasználó által kiválasztott alkalmazási program meghatározott időintervallum során jelentkező legnagyobb adategység átviteli igényét, a legnagyobb igényelt végrehajtási sebességet és a második processzor által igényelt legnagyobb memóriamennyiséget magában foglaló hozzáférést megvalósító eszközei vannak.

Ugyancsak előnyös a találmány értelmében, ha az adatátviteli kérelmeket a memória-forráscím elérésével, az ott lévő adatoknak a puffertár memóriába való átmozgatásával, a memóriacélcím elérésével, valamint a puffertárból az abban lévő adatoknak a célcímre való mozgathatóságát végrehajtó DMA vezérlő- és arbiterpuffertár-memóriát tartalmaz.

A kitűzött feladatot továbbá olyan eljárással oldottuk meg, amely felhasználóalkalmazás-programokat végrehajtó első processzort és a felhasználó alkalmazási programoknak az első processzorban való végrehajtásának támogatása során digitális jelfeldolgozási kérélmeket végrehajtó második processzort tartalmazó multimédia-számítógéprendszer működésének vezérlésére alkalmas. A továbbfejlesztés értelmében a felhasználó által kiválasztott aktív alkalmazási programok végrehajtásának támogatására meghatározzuk a második processzor teljes erőforrásigényét, a második processzor ezen teljes erőforrásigényét összehasonlítjuk a második processzor legnagyobb rendelkezésre álló erőforráskapacitásával, és ha a felhasználóalkalmazás-programok erőforrásigénye következtében a második processzor teljes erőforrásigénye meghaladja a második processzor legnagyobb rendelkezésre álló erőforráskapacitását, akkor megtiltjuk az első processzor által a kiválasztott felhasználóalkalmazás-program végrehajtását, továbbá a második processzorral vett digitális jelfeldolgozási kérélmek függvényében DMA adatátviteli kérelem particionált várakozósort hozunk létre, és a várakozósorba szabályosan ismétlődő időközönként particiójelöléseket helyezünk el.

A találmány szerinti eljárás egy további előnyös fogantatosítási módja értelmében az első processzorhoz vagy a második processzorhoz továbbítandó adatátvitelikérelem-adatokat átmeneti puffertár-memóriába mozgatjuk, majd abból mozgatjuk át az első processzor vagy a második processzor memóriacélcímére.

Előnyös a találmány értelmében továbbá, ha a várakozósor particiójában lévő adatátviteli kérélmeket olyan időközön belül hajtjuk végre, amelyben a második processzorral a várakozósor egy másik particióját hozzuk létre.

Az eddig leírtakat és a találmány lényegét az alábbiakban a csatolt rajz segítségével ismertetjük részletesebben, amelyen a találmány szerinti multimédia-számítógéprendszer és a javasolt eljárás példakénti kiviteli alakját, illetve fogantatosítási módját tüntettük fel. A rajzon az

1. ábra a találmány szerinti multimédia-számítógéprendszer egy lehetséges kiviteli alakjában az első processzor és a második processzor közötti adatáramlást vázolja olyan DMA vezérlőn és arbiteren át, amely mind az első processzort tartalmazó gazdarendszerben, mind a második processzorrendszerben meglévő rendszerszintű cím- és adatbuszokkal kapcsolatban áll, a
2. ábrán egy, egy második processzorrendszer és egy gazdarendszer közötti DMA vezérlő és arbiter révén megvalósított kapcsolat elvi vázolata látható, a
3. ábra az adatáramlást mutatja vázlatosan egy DMA vezérlő- és arbiteregység programozható kiviteli alakjában, a
4. ábra a második processzor által felépített DMA csomagkérelmek formáját és tartalmát mutatja be, valamint azt, hogy ezek a kérélmek mit jelentenek, az

5A. és 5B. ábra a DMA vezérlő és arbiterben lévő adatáramcsomag-pufferek és -regiszterek bejövő és kimenő adatforgalmát mutatja, a

6. ábrán az első processzor által végzett azon művelet elvi folyamatábrája látható, melynek révén felépíti a digitálisjel feldolgozófeladat-listákat vagy -kérélmeket, és azokat úgy kezeli, hogy ne haladják meg a második processzor rendelkezésre álló jelfeldolgozó erőforrását, a

7A., 7B. ábra a DMA vezérlő és arbiter működésének folyamatábráját mutatják a második processzor által épített DMA csomagkérelem-lista feldolgozása során, a

8. ábrán a találmány szerinti multimédia-számítógéprendszer egy előnyös kiviteli alakjának DMA vezérlő- és arbiteregysége látható, a

9A., 9B. ábra a DMA vezérlő és arbiter által végrehajtott DMA átviteli folyamat folyamatábrái, és a

10. ábrán a második processzorban lejátszódó csomagkérelemlista-készítési művelet magas szintű folyamatábrája látható.

A találmányt részletesebben egy csupán példaként bemutatott előnyös kiviteli alak kapcsán ismertetjük a megértéshez és megvalósításhoz szükséges mélységben. Ezen túlmenően jelezzük a vázolt esetben létrejövő problémákat, valamint bemutatjuk a találmány szerinti rendszer és DMA csomagvezérlő mechanizmus használatával elért javulást és rendszer hatékonyságának a növekedését.

A második DSP processzor felé irányuló és az a felől érkező adatáramlás kezelése – mint korábban kifejtettük – potenciálisan szinte megoldhatatlan terheket ró a második DSP processzorra. Például vegyünk egy olyan kereskedelmi forgalomban kapható, elterjedt és ismert számítógépet, amely (számos más személyi számítógép lelkét is alkotó) Motorola 68030 gyártmányú és típusjelzésű processzort tartalmaz. Egy második DSP processzor pedig például a Motorola 56000 gyártmányú és típusjelzésű processzor lehet, amely általában az alaplapra van ráépítve. A 68030 és az 56000 jelű processzor között ugyan létezik DMA útvonal, azonban a DMA vezérlő és arbiter csupán a 68030 jelű első, gazdaprocesszorra támaszkodik, és nem tesz semmit a második DSP processzor működésének megjavítása érdekében. Minden egyes szó (word) esetében, amelyet továbbítani kell a 68030 jelű gazdaprocesszor és az 56000 jelű második DSP processzor között, megszakítást kell a 68030 jelű gazdaprocesszorra bocsátanunk. A megszakítás alatt a gazdaprocesszor busza rákapcsolódik a DMA vezérlő és arbiterre, amely ezt az adat- vagy jelmintaszót továbbítja. Ha ezt a szót a második DSP processzor is kéri, akkor a második DSP processzort is meg kell szakítani. Ez a második DSP processzor leállítja a pillanatnyilag végzett tevékenységét, és az adat- vagy jelmintaszót egy megszakításkezelő révén betölti memóriájába.

A leírt művelet elvégzéséhez szükséges második DSP processzor ciklusainak száma megközelíti a tízet.

Hozzávetőlegesen egy ciklusvesztésig keletkezik a második DSP processzor háromfázisú pipeline szervezése következtében, két vagy három további processzorciklus szükséges a kisszámú regiszter tartalmának elmentéséhez, néhány processzorciklus szükséges továbbá a második DSP processzor memóriája és a modulo számlálóindex vezérlőindex-mutatójának létrehozásához. Végül néhány járulékos processzorciklus szükséges a második DSP processzor állapotának helyreállításához és a megszakított feladathoz való visszatéréséhez. Ez a körülbelül tíz processzorciklus mindössze egyetlen adat- vagy jelmintaszónak a második DSP processzor felé, vagy a felől végzett mozgatásához vált szükségessé. Ez a második DSP processzor a korábban már említett US 4 794 517 számú szabadalmi leírásban részletesen ismertetett ugyancsak háromfázisú pipeline architektúrájú második DSP processzortípusnak felel meg.

Jelen találmányunk olyan javított tulajdonságú multimédiarendszere vonatkozik, amely egy DMA csomagrendszert, azaz olyan programozott mechanizmust használ, amellyel az adatátvitelt lényegesen hatékonyabban tudjuk megoldani. A DMA átviteleket a második DSP processzor által kiadott szoftveres kérelmekkel inicializáljuk, nem pedig a DMA vezérlő és arbiter által kiadott kérelmekkel. A DMA vezérlő és arbiter a második DSP processzorral biztonságosan tud kommunikálni egy olyan, ismétlődő időablakon belül, amelynek hossza igazodik az első processzor többes feladatmenüjében lévő legnagyobb erőforrás-igényű feladathoz. Például egy 726  $\mu$ s hosszú időablak átlagban minden 726  $\mu$ s-ban felveszi a jellemzően 88 200 bit digitális sztereó audiominták lejátszásához szükséges 32 bites (teljes szavas) mintákat. A második DSP processzor jellemző működési sebességén több mint száz ilyen 32 bites csomagot tudna feldolgozni a 726  $\mu$ s időközön belül. Ez elegendő kapacitást jelentene több mint 100 különálló tranzakció vagy kommunikációs csatorna számára, a konkurens jelfeldolgozó rendszerben lévő feladatok és a második DSP processzor alrendszerében végzett műveletek közötti adatáramlásra.

Ha a DMA vezérlő és arbiter kiolvassza egy, a kérelmező második DSP processzor feladata végrehajtása során a particionált várakozósorban elhelyezett csomagátvitel-kérelmet, a DMA vezérlő és arbiter fog döntést hozni a gazdarendszer buszára, például az IBM cég PS/2 számítógéprendszerének Microchannel vagy azzal egyenértékű buszára vonatkozóan, és ha a gazdarendszer engedélyezi a buszhozzáférést a DMA vezérlő és arbiter számára a második DSP processzor által megtekintett kérelmet illetően, akkor a puffertár méretétől függően legfeljebb 16 bájtnyi adatot mozgathatunk át a DMA vezérlő és arbiterben lévő első FIFO (first in first out) regisztertárolóba. Ha a bájtok a FIFO regisztertárolóban vannak, a DMA vezérlő és arbiter veszi át a döntési szerepet a helyi második DSP processzor adatbuszára vonatkozóan, és engedélyezés esetén át fog mozgatni egy 16 bites szót, majd ejti a buszkérelmet. Ezt a szót aztán továbbítja a második DSP processzor saját tárolójába, és a második DSP processzor egy buszciklus idejéig megáll. Így minden egyes továbbított szó esetében az el-

járás egyetlen ciklust vesz igénybe a második DSP processzorban. A DMA vezérlő és arbiter folytatja a döntési tevékenységet a helyi második DSP processzor buszára vonatkozóan mindaddig, amíg a DMA vezérlő és arbiterben lévő FIFO regisztertároló kiürül. A DMA vezérlő és arbiter ezután újabb rendszert generál a gazdarendszer busza felé a továbbítandó többi adatért.

Célszerűen a rendszerebuszra vagy a második DSP processzor buszára vonatkozó döntési folyamatot leghatékonyabban erre a célra kifejlesztett hardverrel és szoftverrel vezényeljük le, amelyek például a DMA vezérlő és arbiterbe implementálhatók. A DMA vezérlő és arbiter csomaglista-feldolgozó képessége hatékony gazdarendszerebusz- és második DSP processzorebusz-használatot tesz lehetővé. A korábban vázolt, ismert megoldás szerinti kialakításra emlékeztetve látható, hogy a javasolt módszerrel átlagosan tíz második DSP processzor ciklusából kilencet meg tudunk takarítani.

Ha például ismét a korábban említett HIFI sztereó jelfeldolgozási feladatot vesszük, amely sztereó CD-zene lejátszásánál jön szóba, úgy egy alkalmazásnak minden másodpercben le kell játszania a szokásos 88 200 16 bites digitális mintát. Ez másodpercenként 88 200 adatszó átvitelét jelenti a gazdarendszer-processzor és a második DSP processzor között, hiszen a gazdarendszer-processzor elolvassa a CD-lemezt, a kiolvasott mintákat átvissza a második DSP processzorhoz, majd a második DSP processzor által feldolgozott audiomintákat fogadja, és azokat analóg integrált hangszegmensekké állítja vissza, másodpercenként 88 200-szor. Az ismert műszaki szint szerinti rendszerben ez a művelet önmagában másodpercenként 88 200 szószor szavanként 10 ciklust, vagy pedig másodpercenként 882 000 végrehajtandó második DSP processzorciklust venne igénybe. A találmány szerinti rendszer előnyös kiviteli alakjánál csupán 88 200 második DSP processzorciklusra, azaz megközelítőleg az ismert rendszer ciklusainak tizedére van csupán szükség.

Az alkalmazott DMA vezérlő és arbiter és listafeldolgozó rendszer megértéséhez képzeljük el, hogy a rendszeren az első processzort utasító felhasználó révén egyidejűleg fut egy beszédfelismerő alkalmazás, valamint egy HIFI sztereó CD-lejátszó-alkalmazás, amelyeknek szükségük van a második DSP processzor támogatási képességeire. A beszédfelismerő alkalmazás megközelítőleg két DMA csatornát vesz igénybe a működéshez, méghozzá egy bejövő- és egy kimenőcsatornát, mindegyiket megközelítőleg 256 szó/csatorna csúcsértékkel. A CD-lejátszó-alkalmazás egyetlen DMA csatornát igényel, körülbelül 64 szó/csatorna csúcsértékkel. Minden alkalommal, amikor a CD-lejátszó-alkalmazás fut, és a minták jelfeldolgozási feladatai a második DSP processzorban futnak, a második DSP processzornak 64 plusz-adatszót kell továbbítania a gazdarendszer-processzor memóriájából a második DSP processzor memóriájába. Ennek elvégzésére a második DSP processzor minden egyes futása alkalmával meghívja a DMA kérelemre vonatkozó működtetőrutinját. Minden egyes meghívás során megadja a következő hozzáférési rendszerprocesszorcímet és a következő belső memóriacímet, és meg-

kezd annak kiszámítását, hogy hol tudja az átvendő 64 szót eltárolni. Minden alkalommal, amikor a beszédfelismerő alkalmazás fut a második DSP processzorban, kétszeresen meg fogja hívni a második DSP processzor DMA kérelmezőrutinját. Mindegyik kérelem olyan egyedi gazdarendszer-rendszercímet fog tartalmazni, ahonnan az információt ki kell olvasni, vagy ahová az információt továbbítani kell, továbbá olyan egyedi második DSP processzormemória-címet fog tartalmazni, ahol a mintákat el lehet tárolni, vagy ahonnan azokat elő lehet venni. A második DSP processzor minden alkalommal, amikor iteratív alkalmazása fut, két-két felismerési sablont fog letölteni, mindegyik sablon megközelítőleg 256 szó hosszúságú.

A három DMA kérelem – a CD-lejátszó-alkalmazás egy DMA kérelme és a beszédfelismerő alkalmazás két DMA kérelme – fellépése során a második DSP processzor révén bekerül egy szegmentált várakozó sorba. Például a beszédfelismerő alkalmazás megkezdődött, és első DMA kérelmét elhelyezte, majd megszakadt, míg a CD-lejátszó-alkalmazás a második DSP processzorban átvette a vezérlést, és elhelyezte saját DMA kérelmét. Miután a CD-lejátszó-alkalmazás DMA kérelmét elhelyezte, a beszédfelismerő alkalmazás újraindulhat és elhelyezheti második DMA kérelmét. Ha csupán ez a három DMA kérelem lett megtéve egyetlen olyan 726  $\mu$ s-os időintervallumban, amelyen belül a második DSP processzor felépít egy particionált várakozósort, akkor azokat a DMA vezérlő és arbiter nem kezeli le az alatt az idő alatt, amíg bekerülnek a várakozó sorba. Az 1378 Hz-es órajel (1/726  $\mu$ s) következő megszakításánál a DMA vezérlő és arbiter megszólítja és feldolgozza a DMA kérelemcsomagok azon következő csoportját, amelyek a megelőző 726  $\mu$ s intervallumban lettek elhelyezve.

A jelen példában a DMA vezérlő és arbiter előbb meg fogja találni azt a csomagkérelmet, amely a beszédfelismerő alkalmazáshoz tartozó 256 szó továbbítására vonatkozik.

Ennek érdekében előbb feltölti a csomagkérelemből a jelzett vezérlőszavakat, hogy megállapítsa, hány szót kell továbbítani, majd betölti a modulo címzésvezérlési határokat. Ezt követően kiolvassa a gazdarendszer memóriacímét, a második DSP processzor memóriacímét és döntést hoz a gazdamedia-buszról. A gazdamedia-buszra vonatkozó engedélyezés vétele után a DMA vezérlő és arbiter a gazdarendszerprocesszor-memóriából saját belső puffertárába előbb áttölt egy bizonyos számú adatbájtot, a bemutatott esetben 16 bájtot. Ezt követően döntést hoz a második DSP processzor buszára vonatkozóan, és minden egyes engedélyezés esetén egy másik szót ír a második DSP processzor memóriájába, a DMA csomagvezérlés-kérelemből kiolvasott második DSP processzor kezdő memóriacímének megfelelően. Miután mind a 16 bájtot beírta, visszatér, és újból döntést hoz a gazdarendszermemória-buszra vonatkozóan. Ez a folyamat mindaddig ismétlődik, amíg mind a 256 szó átvitele megtörténik. Ezután a DMA vezérlő és arbiter kiolvassa a következő csomagot a DMA csomagkérelem-listából. A bemutatott példa szerint ez a

CD-lejátszó-alkalmazás. Ennél a csomagnál 64 szót kell mozgatni, az előbb leírt művelet sor felhasználásával. Miután mind a 64 szó átvitele megtörtént, a DMA vezérlő és arbiter megszólítja a következő csomagkérelmet, amely a beszédfelismerő alkalmazás második csomagja, és miután azt ugyanolyan módon végrehajtotta, mint azt előzőleg leírtuk, nem lát több csomagot, és várakozási állapotba (wait state) kerül a 726  $\mu$ s hosszúságú időablak leletéig. Ha a második DSP processzor által az aktuális 726  $\mu$ s hosszúságú időablak során felépített particionált várakozó sorba több DMA csomagkérelem érkezett, azok lekezelése nem történik meg a következő időablak kezdetéig.

Átérve konkrétan az 1. ábrára, a találmány szerinti eljárás 1 lépésében a gazdagép felhasználója olyan felhasználói programot indít el, amely futása során feltétlenül igényli a digitális jelfeldolgozási feladatok támogatását. Jó példa lehet erre a HIFI sztereó hangvisszaadás, a beszédfelismerés, a modem adatátvitel és a telefax adat-átvitel, a mozgókép lejátszás, a beszéd szintetizálás, vagy a szakember számára jól ismert multimédia környezetbe illő divatos alkalmazások bármelyike. A gazdagép a második DSP processzor feladatai végrehajtását a feladatok azonosítóinak a második DSP processzor operációs rendszeréhez továbbítása révén kérelmezi a 2 lépésben. 3. lépésben a második DSP processzor operációs rendszere bármely aktív feladat kérelméből particionált csomaglistákat épít fel, amelyeket ismétlődő időintervallumonként lezár. A később ismertetésre kerülő 10. ábra a második DSP processzor eme csomaglista-képzési műveletének magas szintű folyamatábráját mutatja be. (Emlékezzünk, a korábban bemutatott előnyös kiviteli alaknál ezek az ismétlődő „csomaglista vége” időközök 726  $\mu$ s-onként történtek.) A második DSP processzor a 4. lépésben látható módon 726  $\mu$ s-os órajellel ütemez egy DMA vezérlő és arbitert, és 111 lépésben folytatja a particionált csomagkérelem-listák felépítését, hogy végrehajthasson bármilyen aktív feladatot, amely vagy a második DSP processzorban fut, vagy pedig egy új felhasználói kérelem alapján a gazdagépből érkezik. 5. lépésben a DMA vezérlő és arbiter veszi a második DSP processzor által 726  $\mu$ s-os időintervallumot kezdő órajelet, 6. lépésben megkezd a csomaglista kiolvasását a második DSP processzor által felépített DMA csomagkérelem-lista egyik particiójából. A DMA vezérlő és arbiter 7 lépésben a rendszer vagy második DSP processzor adatai vagy memóriabuszai vonatkozásában megfelelő döntést fog hozni a kérelmezett csomagátvitelről, 8. lépésben engedélyt kap a megfelelő buszhozzáférésre, majd 9. lépésben adott számú csomagot áttölt a DMA puffertárába, vagy kivesz abból, és amíg az összes átvitel be nem fejeződik a 10. lépésben, addig újra dönt, a rendszer vagy a második DSP processzor busza hozzáférése vonatkozásában, amit a 9 és a 7 lépés közötti visszacsatolással jeleztünk.

Mint az előző rövid vázlatból kitűnt, szükség van némi támogatásra a gazdarendszer részéről (vagy, ha elegendő ehhez a kapacitása, a második DSP processzor részéről, vagy szükség esetén egy további kiegészí-



tőprocesszor részéről) úgy, hogy a második DSP processzor feladatkérelmei elvégezhetőek legyenek a minimális, például 726µs időtartamú időközön belül, vagy pedig számot kell vetnünk annak a következményével, hogy az összes rákövetkező felhasználói feladat igényeit nem vagyunk képesek szorosan valós időben kielégíteni. Ennek a követelménynek a teljesítése érdekében a bemutatott kiviteli alakot egy olyan második DSP processzor erőforrását kezelő és allokáló egységgel látjuk el, amely célszerűen a gazdaprocesszorban foglal helyet. A 6. ábrán vázlatosan bemutatott erőforrás-kezelő és -allokáló rutint tehát a második DSP processzor – vagy szükség esetén más kiegészítőprocesszor – hajtja végre az alábbiak szerint:

Az erőforrás-kezelő és -allokáló rutin figyelemmel kíséri a felhasználó által kiváltott vagy elindított feladatok révén a második DSP processzor számára jelentkező összes terhelést. Ezt a terhelést a teljes DMA bájtvitelti sáv szélességgel, a csomaglista hosszúságával, és a rendelkezésre álló második DSP processzor erőforrás-teljesítményével vagy sebességével definiáljuk, és MIPS-ben (millions of instructions/second, millió utasítás másodpercenként) mérjük a második DSP processzor által végrehajtott ciklusonként. A kezelő- és allokálórutin biztosítja, hogy az összes igényelt második DSP processzorfeladat számára minden egyes 726 µs-os időintervallumban kielégítő gazdaprocesszor-erőforrás áll rendelkezésre, és garantálja minden egyes második DSP processzorfeladat valós idejű DMA igényének kielégítését.

A teljes rendelkezésre álló erőforrás az adott rendszer megvalósításának függvénye, például a második DSP processzor MIPS-ben mért sebességétől, a DMA hardver sáv szélesség-átviteli képességétől, és a második DSP processzor által felépítendő csomagkérelemlista partíciójának hosszától függ. Míg ezek az adott megvalósítás szerint kivétel nélkül változók lehetnek, megvalósítás után az adott rendszerre állandó jellemzők. A DMA bájtvitelti sáv szélességet a gazdaprocesszor-busz sáv szélessége, a második DSP processzor utasításciklus-ideje, vagy a DMA vezérlő és arbiter sáv szélessége közül a legkisebb értékű határozza be. A csomaglistaméretet, mint korábban említettük, a rendelkezésre álló második DSP processzor adattárolási és utasítási órásebessége határozza be.

Az erőforrás-kezelő és -allokáló rutin megvalósításához a gazdarendszer minden egyes felhasználói feladatának tartalmaznia kell valamilyen utalást vagy pontos adatot arról a teljes második DSP processzorfeladat-erőforrásról, amelyet igénybe kíván majd venni, maximális DMA sáv szélesség, csomaglista-hosszúság, és második DSP processzor MIPS-értékekben, amelyeket a szóban forgó feladat maximális esetben igénybe vesz. Mivel a második DSP processzorfeladatokat a gazdarendszernél lévő végfelhasználók veszik igénybe, a 6. ábrán bemutatott erőforrás-kezelő és -allokáló rutin lefoglalja a deklarált erőforrásigényeket a második DSP processzorban lévő kért feladatok céljára. Mindaddig, amíg az összes második DSP processzorfeladat-erőforrás-igény kielégíthető a rendszer keretein belül, a

második DSP processzor feladatai a második DSP processzorba fognak betöltődni. Ezt a gazdarendszer végzi úgy, hogy második DSP processzorfeladat-kérelmet terjeszt be a második DSP processzor operációs rendszeréhez. Ha nem áll rendelkezésre elegendő erőforrás, a felhasználó által beterjesztett második DSP processzorfeladat-kérelmet visszautasítja, és ennek megfelelő felhasználói hibajelzés jelenik meg a gazdarendszerben.

A 6. ábrán látható folyamatábra értelmében a folyamatot a felhasználó indítja úgy, hogy a személyi számítógépén megindítja a 15 lépésben kiválasztott alkalmazást, például beszédfelismerő alkalmazást. A gazdagép meghívja a beszédfelismerő programot, amely megfelelő paramétereket tartalmaz a futásához szükséges igényekre vonatkozóan, nevezetesen a DMA ablakidőre eső maximális szavak számát, a szükséges maximális második DSP processzor MIPS-et, és a második DSP processzor maximális memóriatároló kapacitását. Ez a 6. ábra 11 lépésében zajlik le. A rendelkezésre álló maximális második DSP processzor-erőforrásokat ugyancsak ismeri a rendszer, mert vagy a felhasználó már korábban megadta, vagy pedig a rajzon nem látható, a 20 DMA vezérlő és arbiterben (lásd 2. ábra) és a második DSP processzorban jelen lévő, hardveresen kódolt rendszerfelismerő regiszterekben megvan. Ezt a tényt a folyamat 12 lépése tartalmazza. 13 lépésben kialakítjuk az összes felhasználóalkalmazás-követelményt, például a teljes DMA igény összegét úgy, hogy az összes pillanatnyilag aktív és bármely újonnan kért feladat igényét összegezzük. Ezt az összeget összehasonlítjuk a maximális teljes DMA szóátviteli (bájtvitelti) képességgel, és a teljes második DSP processzor-erőforrással. Ha a szóátviteli teljes igény meghaladja a rendelkezésre álló erőforrás nagyságát, a feladatot a 14 lépésben nem töltjük be, és visszatérünk a 15 lépéshez, azaz a felhasználó által a folytatni kívánt alkalmazások kiválasztásához. Ha a felhasználó által kiválasztott alkalmazások teljes DMA igénye nem haladja meg a lehetőségeket, 16 lépésben összehasonlítjuk az összes aktív felhasználói alkalmazás teljes második DSP processzor utasítás-végrehajtási erőforrásának összegét a maximálisan rendelkezésre álló második DSP processzor MIPS-ei értékével. Ha túlléptük a maximumot, az új alkalmazást nem töltjük be, azaz visszatérünk a 14, majd 15 lépésre. Ha azonban a teljes rendelkezésre álló második DSP processzor MIPS-értéket az említett összeg nem haladja meg, az eljárást 17 lépésben folytatjuk, ahol létrehozuk az összes aktív felhasználói alkalmazás teljes második DSP processzor tárolási igénye összegét, majd ezt összehasonlítjuk a maximális rendelkezésre álló második DSP processzor adattároló kapacitása értékével. Ha annak maximumát az összeg értéke nem haladja meg, a második DSP processzor megindítja a (akár a második DSP processzorban, akár szükség esetén egy segédprocesszorban található) lekezelési műveletet, amelynek során elvégzi a személyi számítógépen az erőforrás-lekezelő és -allokáló rutint, majd 18 lépésben betölti az alkalmazást, egyben jelzi operációs rendszerének, hogy kezdjen hozzá az újonnan kiválasztott felhasználói alkalmazás végrehajtásának, majd az eljárást 19 lépésben befejezzük.



Mint korábban említettük, a DMA csomagkérelem-listákat particionált formában a második DSP processzor memóriájában saját operációs rendszere hozza létre. Mivel a második DSP processzor alkalmazáskérelmei bekerültek a második DSP processzor operációs rendszerébe, az elkészít egy, a kívánt alkalmazások végrehajtásának támogatásához szükséges DMA csomagkérelem-listát. A DMA csomagkérelmek lehetséges alakját a 4. ábrán tüntettük fel. A 4. ábrán ötszavas DMA csomagkérelem látható, amely első és második vezérlőregiszterekben tárolt két vezérlőszót, a gazdarendszermemória-cím alsó címét és a felső címét jelentő két szavát, valamint a második DSP processzor memóriacímre egy szavát tartalmazza, mely azt a helyet jelöli, ahová, vagy ahonnan a szavakat továbbítani kell. Ennek a vezérlőregiszterekhez tartozó speciális vezérlőszónak a kódolását követhetjük nyomon a 4. ábrán. A vezérlőszavakat a 20 DMA vezérlő és arbiter használja fel a későbbiekben részletesebben ismertetendő módon.

A második DSP processzor operációs rendszere felépíti a memóriában a DMA csomagkérelem-listát. A címzés olyan, hogy a memória a második DSP processzoron belül visszatérő puffertárként működik. A csomagkérelem-lista olyan particionált lista, amelyben egy vagy több egyedi DMA csomagkérelem-csoport, és a végén „várakozóállapot” vagy „lista vége” jelzés, azaz marker található. A „lista vége” vezérlő jelzőbit a második DSP processzor memóriájában olyan mutatót tartalmaz, amely visszamutat a kezdetre úgy, hogy a puffertár végtelen ciklusban működik. A puffertár létrehozására rendelkezésre álló második DSP processzor memóriájának kellőképpen nagyra kell lennie ahhoz, hogy bármelyik időpillanatban legalább két teljes csomagkérelem-listát tudjon befogadni. Ez azért lényeges, mert a 20 DMA vezérlő és arbiter hardvere az egyik kérelem-lista tartalmának feldolgozását végzi, míg a második DSP processzor operációs rendszere a lista következő partíciója tartalmának létrehozása miatt foglalt. A második DSP processzor operációs rendszere az adatok továbbítása céljából úgy hozza létre a DMA csomagkérelem-listákat, hogy a csomagkérelmeket egymás után hozzacsatolja az aktuális csomagkérelem-lista-tartalmához, ahogyan a második DSP processzorban az aktív alkalmazások szükség szerint vagy új jelminták feldolgozás céljából történő bevételezésére, vagy a már feldolgozott jelmintáknak a kérelmező felhasználói alkalmazáshoz való visszaküldésére megteszik DMA kérelmüket.

Mint a 10. ábrán látható, a második DSP processzorban futó adott alkalmazás 61 belépési pontban egy DMA csomagárvitelt oly módon kérelmez, hogy először 62 lépésben betölt egy csomagmutatót, 63 lépésben a csomagkérelmet beírja a csomagkérelem-listába, azután 64 lépésben megnöveli a csomagmutatót, hogy az a következő csomag helyére mutasson, majd 65 lépésben kiolvassa az adatokat a csomagmutató által jelzett helyről. 66 lépésben ellenőrzi, hogy a csomagmutató a csomagkérelem-lista végére mutat-e, és ha igen, akkor 67 lépésben a csomagmutatót beállítja arra a helyre, ahol a csomagkérelem-lista vége jelölést közvetlenül követő csomagkérelem listakezdet mutatója áll,

majd 68 lépésben a csomagmutatót elmenti, és 69 kilépési pontban visszatér az eredeti alkalmazás futtatásához. Ha a csomagmutató nem a csomagkérelem-lista végére mutat, akkor a 67 lépés kimarad a vázolt folyamatból. A második DSP processzoralkalmazások bármikor tehetnek csomagkérelmeket. Ilyen kérelmek aszinkron kapcsolatban állhatnak a 20 DMA vezérlő- és arbiter-lista-végrehajtással, amelyet a 20 DMA vezérlő és arbiter vezényel le. A 20 DMA vezérlő és arbiter által végrehajtott csomagkérelem-lista a második DSP processzor operációs rendszere által precízen meghatározott időintervallumokra van felosztva, amely az éppen felépítés alatt álló csomagkérelem-listában a partíció végére „csomagvárás” vagy „lista vége” jelet helyez el. A bemutatott előnyös kiviteli alaknál ezek a jelölések előre meg vannak írva a memória várakozósorában úgy, hogy azok szabályos időközönként bekövetkeznek, amint a várakozósort a 20 DMA vezérlő és arbiter olvassa. Pontosan visszatérő időközönként a második DSP processzor jelzi a 20 DMA vezérlő és arbiternek, hogy kezdje meg az éppen soron következő csomagkérelem-lista végrehajtását. A „várakozó állapot” („lista vége”) jelölés a 20 DMA vezérlő és arbiter leállítására szolgáló jelölésként hat, ha az befejezte az aktuális lista feldolgozását. Mivel a második DSP processzorban az alkalmazások folyamatosan megteszik a DMA csomagátviteli kérelmeket, az operációs rendszer megkezd a csomagkérelem-listában a következő partíció betöltését. A második DSP processzorok, mint például a már korábban említett Motorola 56000 típusú processzor, az iparban jól ismertek, és operációs rendszerük kapacitása, képessége is ismert, amellyel ilyen listákat tudnak a memóriában felépíteni, ezért szakember számára további ismertetés feleslegesnek tűnik.

A 2. ábrán egy tipikus gazdarendszertől, annak adat- és címbuszain át egy 20 DMA vezérlő és arbiterhez irányuló információcsomagok általános fizikai kialakítása és adatforgalma látható. Egy tipikus gazdarendszer például az IBM PS/2 típusú számítógépcsaládjának valamelyik tagja, vagy pedig a korábban említett Motorola 68030 processzort tartalmazó számítógép lehet, a 20 DMA vezérlő és arbiter pedig például az Intel cég 82325 típusjelzésű programozható mikrocsatornás (Microchannel) 20 DMA vezérlő és arbiter lehet. Ugyancsak láthatók a 20 DMA vezérlő és arbiter-től egy jellemző második DSP processzorhoz irányuló interfészek. Tipikus második DSP processzor például az IBM cég US 4 794 517 számú szabadalmi leírásából megismerhető háromfázisú pipeline szervezésű második DSP processzor, vagy a korábban ismertetett Motorola 56000 típusú processzor. Az adatcsomagok a gazdarendszer memóriája és a 20 DMA vezérlő és arbiteren belüli puffertár között és a puffertár és a második DSP processzor adatokat vagy utasításokat tartalmazó memóriája között buszokon át haladnak.

Egy programozható 20 DMA vezérlő és arbiter, mint például a kereskedelmi forgalomban kapható Intel 82325 csipkészlet, eredményesen alkalmazható ezekre a célokra. Ezt leírásunkban „buszmesterhardver”-ként is említhetjük, amely a gazdagép adattároló részegység-

ge vagy memóriája, és a második DSP processzor utasítás- vagy adattároló eszköze közötti kérelmezett adat-csomagok aktuális átvitelét végzi. A buszmesterhardver két fő funkcionális egységre bontható: az egyik a csomagkérelemlista-processzor, a másik a DMA átvitelkezelő. A csomagkérelemlista-processzor a második DSP processzor operációs rendszerétől „start” léptetőjelet kap egy például 726  $\mu$ s hosszúságú időköz kezdetén, ennek hatására megkezd a második DSP processzor memóriájában felépített csomagkérelemlista egyik partíciójának olvasását attól a címtől; ahol legutoljára partíció határjelző szerepet betöltő „csomagra várás” jelölést fedezett fel, majd a második DSP processzor memóriájából származó DMA csomagkérelemlében lévő vezérlő- és címinformációt feldolgozza és továbbítja a DMA átvitelkezelőnek, amely elvégzi a meghatározott DMA átvitelt.

A csomagkérelemlistában lévő vezérlő- és címinformációt feldolgozás után továbbítja a 20 DMA vezérlő és arbiter átvitelkezelőjének, amely a meghatározott DMA átvitelt bármely más normális buszmester módon meghajtott eszközhöz hasonlóan elvégzi. A folyamat ezt követően folyamatosan egymás után csomagokat kér mindaddig, amíg ismét „csomagra várás” jelölést észlel. Ez leállítja a buszmesterhardvert és megszakítja a csomagkérelemlista-partíció feldolgozását.

A korábban felhozott példához visszatérve, amelyben a felhasználó egyidejűleg kíván személyi számítógépén beszédfelismerő alkalmazást és HIFI sztereó CD-lejátszást egy második DSP processzor segítségével végrehajtani, a felhasználó a beszédfelismerő programot és a HIFI sztereó CD-lejátszó-programot például a személyi számítógépéhez csatlakozó egérrel vagy kurzorral, a programhoz tartozó ikon kiválasztásával előbb kiválasztja. Ez a kiválasztás megindít egy kérelmet a megfelelő második DSP processzor alkalmazásai betöltésére. A kérelem előbb végighalad a 6. ábrán bemutatott folyamaton, hogy egyrészt megbizonyosodjon, hogy megfelelő mennyiségű második DSP processzor-erőforrás és DMA sáv szélesség áll rendelkezésre. Mielőtt a kiválasztott alkalmazásokat aktuálisan betölti, munkába állnak az erőforrás-kezelő és -allokáló rutinok (a bemutatott előnyös kiviteli alaknál például a gazdagépen), hogy megbizonyosodjanak, hogy elegendő erőforrás áll rendelkezésre a felhasználónak a kiválasztott alkalmazások meghívása révén kinyilvánított második DSP processzoralkalmazás-követelmények kielégítésére. Ha elegendő erőforrás létezik, akkor az alkalmazások betöltődnek és a funkciók működni kezdenek a második DSP processzorban. Egyébként a rendszer megfelelő hibajelzést generál a felhasználó gazdagépén, hogy ezen keresztül jelezze: a kiválasztott alkalmazások nem töltődtek be.

Ismertetésünk során feltételezzük, hogy a második DSP processzor operációs rendszere már korábban inicializálásra került, és hogy bármely újonnan kérelmezett második DSP processzorfunkció fogadását megelőzően a csomagkérelemlista-puffertárban lévő bármely korábban kérelmezett funkcióhoz megfelelő módon hozzá lehetek rendelve a DMA átviteli kérelmek és jelölések, és a

20 DMA vezérlő és arbiter hardverét minden egyes partíciót záró, például a jelzett 726  $\mu$ s-os időintervallumban inicializáltuk. Feltételezve, hogy nem volt korábbi feladat vagy alkalmazás kérelmezve, ez a példa azt mutatja, hogy a beszédfelismerő alkalmazás két DMA csatornára, valamint 256 szó/csatorna csúcscsértékű sáv szélességre tart igényt, míg a HIFI sztereó CD-lejátszó alkalmazás egy DMA csatornára és 64 szó/csatorna csúcscsértékű sáv szélességre tart igényt. Minden alkalommal, amikor a CD-lejátszó-alkalmazás a második DSP processzorban fut, a gazda-rendszerprocesszor memóriájából a CD-lemezzel eredetileg kiolvasott digitális jelminiták alakjában 64 újabb szót kell továbbítani. Ezeket a szavakat a gazda-rendszermemóriából a második DSP processzor memóriájába, pontosabban a körforgó csomagkérelemlista-puffertárba kell végrehajtás céljából áttenni. Ehhez a második DSP processzor meghívja az operációs rendszere rutinját DMA kérelem megtételére, és ezt minden alkalommal megteszi, amíg az alkalmazás fut, azaz minden alkalommal, amikor 64 szavas mintát kell feldolgozni. Minden egyes meghívás során a második DSP processzor megadja az írandó vagy a kiolvasandó következő gazdarendszerprocesszormemória-címet és a következő második DSP processzorpuffertár-címet, amelyen a 64 szó számolása megkezdődik.

Ezen túlmenően minden alkalommal, amikor a második DSP processzorban a beszédfelismerő alkalmazás fut, meghívja a második DSP processzor operációs rendszerének DMA kérelmet létrehozó rutinját. Minden egyes DMA kérelem egyedi rendszercímet és egyedi második DSP processzorpuffertár-címet tartalmaz, amelyre a gazdarendszer minden egyes futása alkalmával a gazdarendszertől két felismerési sablont tölt le. Mindegyik sablon 256 szó hosszúságú a legtöbb beszédfelismerő programban, így mindent összevetve a DMA csomagkérelemlistába a második DSP processzor operációs rendszerével egymást követően három DMA kérelmet kell írni, amikor időszertűvé válik.

A három DMA csomagkérelmet a 20 DMA vezérlő és arbiterhardver fogja lekezelni a következő 726  $\mu$ s-os időtartam alatt, a partícióban lévő jelöléstől kezdve. Ez a jelölés, mint említettük, vagy egy „várakozóállapot” vagy „lista vége” jelölés a második DSP processzor által létrehozott DMA csomagkérelemlista-szegmensben. Ennek a következő időtartamnak a kezdetén a második DSP processzor operációs rendszere „várakozó állapot” csomagkérelmet helyez el végjelölésként az éppen létrehozott csomagkérelemlista-partícióban, és egy órajellel vagy léptetőparanccsal indítja a buszmesterhardvert.

A buszmesterhardver a második DSP processzor memóriájában megtalálja azt a csomagot, amely kérelmezi a beszédfelismerő alkalmazáshoz a 256 szó mozgatóját, majd feltölti a csomagkérelemlében jelen lévő vezérlőszót, hogy megnézze, hány szót kell mozgatnia, és hogy megállapítsa a modulo címvezérlő határokat. Ezután kiolvassa a gazdarendszerprocesszormemória- és a helyi második DSP processzormemória-címeket a DMA csomagkérelemléből. Ezt követően döntést hoz a személyi számítógép mikrocsatornás, vagy más típusú gazdabuszára vonatkozóan, és ha hozzáférést kap, néhány

(16 bájt) adatot áttölt a gazdaprocesszor memóriájából a buszmasterhardverben lévő puffertárba. A 20 DMA vezérlő és arbiter ezután döntést hoz a második DSP processzor memóriabuszhoz, és minden alkalommal, amikor engedélyezést kap, másik szót ír a második DSP processzormemóriába annak a második DSP processzor-memória-címnek megfelelően, amelyet a DMA csomagkérelem-listából kiolvasott. Ez a folyamat a korábban leírtak szerint mindaddig folytatódik, amíg megtörténik az összes szükséges DMA csomag feldolgozása. Abban az esetben, ha a második DSP processzor operációs rendszere több DMA csomagkérelmet helyezett el az aktuális 726  $\mu$ s időintervallumban, nem történik meg a többletkérelmek lekezelése az adott időintervallumban, hanem csak a következő időköz kezdetekor. A folyamat mindaddig ismétlődik, amíg a gazdarendszer felhasználója befejezi az összes, a második DSP processzort igénybe vevő alkalmazást. A bemutatott elrendezésből kitűnik, hogy a DMA csomagkérelem-lista a második DSP processzorral és a DMA sáv szélesség-allokációval és -vezérlő folyamattal (amelyet a bemutatott kiviteli alaknál a gazdagép vezényel le) garantálja, hogy az összes DMA csomagkérelem két alapórjel-intervallumon belül átkerül a 20 DMA vezérlő és arbiterhez. Így a második DSP processzor operációs rendszere által a memóriájában felépített particionált listában lévő függő DMA kérelmeket a DMA hardver egyetlen időközön belül megszólítja és kiolvassa, és a következő időintervallumban, a bemutatott esetben 726  $\mu$ s időtartamon belül végrehajtja. Ha a jelmintákat a gazdagép memóriájából vételezzük és a második DSP processzor memóriájába továbbítjuk, akkor ez 726  $\mu$ s időtartamon belül történik meg. A rá következő időtartam során a második DSP processzor megkezd az adatminták feldolgozását és új DMA csomagkérelmek létrehozását a feldolgozott jelmintáknak a gazdarendszerhez való visszaküldése érdekében. Ezek a kérelmek a 20 DMA vezérlő és arbiter által éppen feldolgozott DMA csomagkérelem-lista-partíciót követő partícióba kerülnek. Így a következő időintervallumban a befejezett jelmintákat a második DSP processzorból DMA átvitelrel vissza tudjuk juttatni a gazdarendszer-processzorhoz, átlagosan csupán két DMA időintervallumnyi kezelési időkéssel.

Ugyanezt a működési sebességet érjük el az összes függő alkalmazáskérelemnél, feltételezve, hogy az alkalmazások által igényelt teljes második DSP processzor-erőforrást és teljes DMA sáv szélességet nem lépjük túl. Ezt a gazdarendszer személyi számítógépén jelen lévő és futó allokáló- és vezérlőmechanizmussal tudjuk biztosítani, ami azt jelenti, hogy a második DSP processzorban futó, szorosan valós idejű alkalmazások garantáltak elegendő mennyiségű adatot tudnak szállítani, vagy kellő mennyiségű feldolgozott adatot tudnak vételezni, hogy kielégítsék bármely kért felhasználói alkalmazás követelményeit. Mivel a gazdarendszerben futó összes alkalmazásnak meg kell határoznia előre, hogy mennyi a csomagkérelem által átvendő maximális szószám, és hogy mennyi a második DSP processzor maximálisan igényelt utasításfeldolgozó MIPS-értéke, az

erőforrás-allokáló és -vezérlő mechanizmus a gazdagépben nagy pontossággal tudja, hogy milyen hosszú egy adott szinten a teljes DMA csomagkérelem-lista, és mikor következik be a kapacitás túllépésének veszélye, ezért nem is fogja engedélyezni új alkalmazások beindítását.

A 20 DMA vezérlő és arbiter így folyamatosan feldolgozza a második DSP processzor memóriájából érkező azon DMA csomagkérelmeket, amelyeket az előző időintervallumban az egyes alkalmazások helyeztek oda, és amíg a 20 DMA vezérlő és arbiter az aktuális DMA csomagkérelem-lista-partícióján dolgozik, a második DSP processzor memóriájába új DMA csomagkérelmek kerülnek a rá következő DMA csomagkérelem-várakozó sor partíciójába, amelyekhez a rá következő időintervallumban a DMA hardver hozzáfér és végre tud hajtani. Ez a kialakítás egyszerű migrációs útvonalat biztosít egy adott rendszerben a második DSP processzorok számának a megtöbbszörözésére. Ilyen esetben mindegyik második DSP processzornak megvan a saját DMA csomagkérelem-végrehajtó hardveregysége, amely a második DSP processzor busza vagy a gazdagépbusz számára buszmasterként szolgálna. Abban az esetben, ha az összes második DSP processzor ugyanazt a megszakításidőzítót órajelet használná, mint a 20 DMA vezérlő és arbiter, akkor ezen túlmenően a gazdarendszer-memória egyszerű eszközöket, vagy puffertárat tudna rendelkezésre bocsátani az egyes második DSP processzorok közötti adattovábbításra.

Látható, hogy a DMA csomagkérelem-végrehajtó 20 DMA vezérlő és arbiter megbízható megoldást nyújt a nagyszámú kommunikációs csatornák, vagy egy gazdaprocesszor és a második DSP processzor, vagy több második DSP processzor közötti kérelmek lekezelésének problémájára, és a különböző processzorok bármelyikében futó alkalmazások futásának a biztosítására. Ha önmagában száznál több DMA csatornát használó hardvereszközt próbálnánk megvalósítani, például száz egyedi DMA hardvereszközt, vagy huszont négy DMA csatornát használó hardvereszközt, ezt olyan drága rendszert eredményezne, amelyet szinte lehetetlen lenne vezérelni és felügyelni. A 20 DMA vezérlő és arbiter és a csomagkérelem-lista-processzor lehetővé teszi, hogy ne legyen szükség száz DMA csatorna közötti egyeztetésre, mivel azok abból a szempontból multiplexeltnek minősülnek, hogy mindegyikük számára elegendő, ha a visszatérő időközön belül az egyedi buszengedélyezés-kérések révén, továbbá olyan engedélyezéssel és adatátvitel-csomagok révén, amelyek az előírt időközön belül garantáltan végrehajthatók, soros módon kerülnek feldolgozásra.

Az eddigiekből világossá válik, hogy a 20 DMA vezérlő és arbiter a kulcseleme a találmány szerinti és a 2. ábrán felvázolt rendszernek. A 3. ábrán egy ilyen 20 DMA vezérlő és arbiter főbb vezérlési adatáramát tüntettük fel. Ilyen 20 DMA vezérlő és arbitert kereskedelmi forgalomban beszerezhetünk, csupán az a lényeg, hogy kielégítő hardver- és szoftverkapacitása legyen a 3. ábrán látható feladatok végrehajtására. Erre a célra például megfelelő az Intel cég 83325 típusjelzésű

programozható buszmester 20 DMA vezérlő és arbiter áramköre. Az egység négy hardvervezérlőt használ a 3. ábrán látható módon, nevezetesen 310 csomagvezérlőt, 311 DMA kezelőt, gazdarendszer-buszmester 312 DMA vezérlő és arbitert, és második DSP processzor buszmester 313 DMA vezérlő és arbitert. A 310 csomagvezérlő a második DSP processzor adatmemóriájában lévő, a második DSP processzor operációs rendszerre által odahelyezett particionált csomagkérelem-listából csomagvezérlő szavakat olvas ki. Az egy adott csomagkérelemhez tartozó és a 4. ábrán feltüntetett öt vezérlőszó kiolvasását követően a 310 csomagvezérlő indítja a 311 DMA kezelőt.

A 311 DMA kezelő a DMA csomagvezérlő szóban lévő csomagbájtyszámot több adatcsomaggá bontja szét. Az információátvitel irányától függően, amelyet a DMA csomagkérelem átviteli bitje jelöl, a 311 DMA kezelő ismételten inicializálja vagy a gazdarendszer-buszmester 312 DMA vezérlő és arbitert, vagy a második DSP processzor buszmester 313 DMA vezérlő és arbitert, hogy döntést kezdeményezzen a buszok vonatkozásában, és hogy áttöltse a megfelelő méretű adatcsomagot arra a kezdőcímrre, vagy arról a kezdőcímről, amely a második DSP processzor memóriájából kiolvasott DMA csomagkérelemben meg lett adva. Ha az érintett 20 DMA vezérlő és arbiter befejezi a DMA átvitelt, azt jelzi a 311 DMA kezelőnek, amely vagy egy új adatcsomagot jelez, vagy a számlálás eredményeképpen felismeri és nyugtázza a befejezést a 310 csomagvezérlő felé. A 310 csomagvezérlő egy másik DMA művelethez tartozó másik csomagvezérlő szókészlet olvasását kezdi meg, és a folyamat folytatódik mindaddig, amíg a második DSP processzor DMA csomagkérelemlista-partícióján végighaladva „várakozási állapot” jelölést fedez fel.

A függő DMA csomagkérelemek a 4. ábrán látható módon öt szót tartalmaznak, amelyek információformátuma a második DSP processzor memóriája kezdőcímeinek, a személyi számítógép felső és alsó tartománya kezdőcímeinek és a DMA processzor vezérlőregiszter-tartalmának megfelelően szét van szedve. Mindegyik ötszavas DMA csomagkérelem a 3. ábrán látható 310 csomagvezérlőbe jut, és a 7A. és 7B. ábrán nyomon követhető módon kerül végrehajtásra a 310 csomagvezérlőben lévő DMA listaprocesszor révén.

A 7A. és 7B. ábrán a „várakozási állapot” olyan DMA csomagkérelemlistapartíció-határ, amelyet a 4. ábrán látható irányvezérlő 01 és 02 iránybitjei rögzítenek. A 4. ábrán mindkét 01, 02 iránybit értéke nulla. Egy várakozási állapot a 3. ábra 310 csomagvezérlőjében végrehajtásra kerülő DMA listafolyamat kezdőpontja. Ez a 21, 22, 23 és 24 vagy 25, 26, 27, 28 lépésen át tart a 7A. ábrán nyomon követhető módon. Az irányvezérlő biteket a 7B. ábra 29 lépésében olvassuk ki, és a 4. ábra irányvezérlő 01 és 02 iránybitjei tartalmával összhangban lévő döntést hozunk, hogy várakozó üzemmódot kezdünk-e meg, vagy hogy megkezdjük az áttöltést a gazdarendszer memóriájából a második DSP processzor memóriájába, vagy a második DSP processzor memóriájából a gazdarendszer memóriá-

jába. 30–32 lépésekben egy mutató értékét növeljük, 33A, 33B és 33C lépésekben pedig a teljes értéket összehasonlítjuk a maximális 256 szónak megfelelő, legfeljebb 255 értékkel.

Az 5A., 5B. ábrákon a 3. ábra 310 csomagvezérlőjében 1 kiképzett néhány aktív puffertárat és regisztert tüntettünk fel vázlatosan. A DMA csomagmutatót alkotó 514 regisztert több szegmens alkotja. Az 514 regiszter olyan 16 bites regiszter, amelyet kizárólag a második DSP processzor tud kiolvasni és a második DSP processzor adatmemóriájában a DMA csomagkérelemlistához tartozó mutató 12 legkisebb értékű bitjét tartalmazza. Az 514 regisztert hardveresen frissítjük fel, és a 3 legnagyobb értékű bit értéke nulla lesz. A 3 legkisebb értékű bit és a 15 bit, mint olyan, nem létezik, és az adott megvalósításnál nulla értékűként tekintjük őket. Az adattárolóban lévő adatokat, amelyeket ebbe az 514 regiszterbe töltünk be, bájtcsímként tároljuk. 519 regiszter több részből tevődik össze, és tartalmazza a rendszer vagy gazdarendszermemória-címet. Ez egy olyan 32 bites regiszter, amelyet sem a második DSP processzor, sem a gazdarendszer processzora nem tud olvasni vagy írni. Ez az 519 regiszter tartalmazza a gazdarendszermemória vagy I/O terület címét, amelyet a csomagátvitel céljára használunk fel. Az 519 regiszter alsó 24 bitje automatikusan megnövekedik a gazdarendszerprocesszormemória-átvitelkor. A memória vagy az I/O terület választását a rendszermemória/bemeneti-kimeneti bit jelzi. Az ebben az 519 regiszterben tárolt cím egy bájt cím.

Az 518 regiszter a második DSP processzormemória-címet tartalmazza, és egy 16 bites önnövelő 518 regiszter, amelyet sem a második DSP processzor, sem a gazdarendszer személyi számítógép nem tud olvasni vagy írni. Az 518 regiszter tartalmazza a csomagátvitelhez felhasznált második DSP processzor adat- vagy utasításmemóriájának a címét. Az 518 regiszter tartalmának növelésére modulo vezérlőbiteket használunk. Az adattárolóban lévő adatokat, amelyeket az 518 regiszterbe kívánunk betölteni, szó- vagy utasításcímként tároljuk, ahol csupán az alsó 15 bitet használjuk fel az adattárolóhoz tartozó cím képzésére. A 3. ábrán látható 311 DMA kezelőt a 310 csomagvezérlőtől érkező indító DMA jel inicializálja. A 311 DMA kezelő először alakítja a 310 csomagvezérlőtől érkező csomagszámot egy csomagbájtsszámmá. A 311 DMA kezelő feladata az, hogy ezt a csomagbájtsszámot löketszámokká bontsa szét, mivel a 20 DMA vezérlő és arbiter belső puffertára két 16 bájtsszor 8 bites RAM-ot tartalmaz a bemutatott kiviteli alaknál, így a löketnagyság egyszerre 32 bájtira korlátozódik. Ha a vezérlőregiszterek bájtja vagy szóbitje bájt üzemmódot mutat, az adat nem csomagolódik be az adattárolóba. Ilyenkor csupán az adattároló al-sórendű bájtját használjuk a buszmesterműveletekhez. Ilyen feltételek között normális, 16 bites löketméretet alkalmazunk.

A DMA átvitel a löketszám felhasználásával megy végbe. A löketszám megegyezik a löketnagysággal, kivéve, ha a maradék vagy kezdő csomagszám kisebb, mint a löketméret. A csomagbájtsszámot ismétlődően

különálló löketekké bontjuk le, mindaddig, amíg el nem fogy.

A 8. ábrán a kiindulási bájtyszámot betöltjük 35 akkumulátorba. A 311 DMA kezelőhardvere tartalmazza a 8. ábrán látható hardvert. A löket nagyság a DMA csomagkérelemben lévő vezérlőinformációba épített bájt/szó bitértékén alapul. Ha ez a jelzőbit nulla értékű, akkor a 35 akkumulátorban lévő érték meghaladja a löket nagyságot. A löket nagyságot a korábban leírt módon használt rendszernek megfelelően előre megválasztjuk. A bájtyszámot betöltjük a rendszerbe és a második DSP processzor DMA bájt számlálójába, és a 311 DMA kezelő figyelni az irányjelző biteket.

Ha a 7B. ábra 29 lépésében a kiolvasott irányjelző bitek a gazdarendszer-memóriából történő kiolvasást jeleznek, a 311 DMA kezelő a bemeneti adatmultiplexer vezérlését a rendszeradatkapuhoz kapcsolja. A rendszer-puffertár-vezérléseket is kiválasztja, és indítja a gazdarendszer-buszmaster 312 DMA vezérlő és arbitert. Az működése során betölti a belső RAM 314 puffertárát, és ezt a műveletet úgy értelmezi, mint egy, a gazdarendszer-memóriától a 314 puffertár felé irányuló olvasást. A 314 puffertárba történő adatbeírás után a gazdarendszer-buszmaster 312 DMA vezérlő és arbiter nyugtázza a 311 DMA kezelő felé megtörtént átvitel befejezését.

A befejezési jel vétele hatására a 311 DMA kezelő ezt követően a puffertárvezérlőket hozzákapsolja a második DSP processzor 313 DMA vezérlő és arbiterehez, amely folytatja az adatok átvitelét a 314 puffertárból a második DSP processzor memóriájába, és annak befejeződése után igazolást küld a 311 DMA kezelőnek, hogy az átvitel befejeződött. Ha a 311 DMA kezelő az átvitel aktuális irányaként a rendszermemóriába beírást észlel, a 311 DMA kezelő a bemeneti adatmultiplexer vezérléseit hozzákapsolja a második DSP processzor DMA adatportjához, és a második DSP processzor DMA puffertárvezérlői hozzárendelődnek a második DSP processzor 313 DMA vezérlő és arbitereinek 314 puffertáraihoz. A 311 DMA kezelő ezután indítja a második DSP processzor 313 DMA vezérlő és arbitert, amely a műveletet mint a második DSP processzor memóriájától a 314 puffertárba irányuló olvasást értékeli. A 313 DMA vezérlő és arbiter folytatja a 314 puffertár feltöltését, és az adatok 314 puffertárba való beolvasása után nyugtázza az átvitel befejeződését a 311 DMA kezelő felé. Ha észleli, hogy a második DSP processzor DMA művelet befejeződött, akkor a 311 DMA kezelő a puffertárvezérlőket átkapcsolja a rendszeroldalra, inicializálva ezzel a gazdarendszer 312 DMA vezérlő és arbitert, és átvive az adatokat a 314 puffertárból a gazdarendszer memóriájába. Befejezés után a 312 DMA vezérlő és arbiter nyugtázza az átvitel befejezésének tényét a 311 DMA kezelőnek.

Miután az első löketet átvittük, a 311 DMA kezelő a 8. ábra 35 akkumulátorát feltölti a maradék bájt számmal. Ha ismételtlen a jelzőbit értéke nulla, a löket számot azonossá teszi a löket nagysággal, és ezt az adatszeget DMA művelettel átvviszi. Ha azonban a jelzőbit értéke egy, a 35 akkumulátorban lévő szám kisebb lesz, mint a 36 löketregiszter löket nagysága. A kettős

37 multiplexer ezt követően kiválasztja a 35 akkumulátor tartalmát az utolsó és végső DMA átvitelhez. Ha a jelzőbit értéke egy, azt negatív számként értékeli, míg ha a jelzőbit értéke nulla, azt pozitív számként vagy nullaként értékeli. A nulladetektálás azért fontos, mert az jelzi, hogy nulla löket számot nem kell végrehajtani. Miután a csomagbájt szám elfogyott, a 311 DMA kezelő a 3. ábra 310 csomagvezérlőjének a befejezés nyugtázásával válaszol. A második DSP processzor 313 DMA vezérlő és arbitere, amelyet a 3. ábrán látunk, része az Intel cég korábban említett 82325 típusú programozható 20 DMA vezérlő és arbitereinek. Ez a második DSP processzor cím-számlálót mind modulo-, mind bájt számlálóként használja, amelyet a 3. ábrán a 315 és 316 vezérlőszótárból a 313 DMA vezérlő és arbiter felé irányuló nyilak értelmében a löket számmal töltünk be. A 310 csomagvezérlő a 313 DMA vezérlő és arbiteren a címszámlálót tölti, míg a 311 DMA kezelő a bájt számlálót tölti, és visszaállítja a belső puffertár címet, amikor indítja a második DSP processzor DMA műveleteket. A 311 DMA kezelő a DMA puffertár adat-útvonalakat is beállítja, és vezérli még a második DSP processzor 313 DMA vezérlő és arbitereinek megindítása előtt. A második DSP processzor DMA bájt számlálóját a 313 DMA vezérlő és arbiterben egyesével vagy kettőssel számlál, ha a buszmaster működése adattárolást is magában foglal. A bájt számláló bájt üzemmódban egyesével, szó üzemmódban kettőssel számlál. A bájt számláló négyesével számlál, ha a művelet utasítástárolást is magában foglal. Startjel vételének hatására a második DSP processzor 313 DMA vezérlő és arbitere működésbe lép, és vagy adatokat olvas ki a második DSP processzorból és tárolja el azokat a belső 314 puffertárba, vagy az adatokat a 314 puffertárból olvassa ki és tárolja el azokat a második DSP processzorba, vagy a 14 puffertárból kiolvasott adatokat a második DSP processzor utasítástárolójába tölti be, a 4. ábrán látható információ 01, 02 iránybitekkel összhangban.

A 3. ábra gazdarendszer-buszmaster 312 DMA vezérlő és arbitere ugyancsak a 82325 típusjelzésű Intel gyártmányú csip részét képezi, és felelős az adatok kétirányú mozgatásáért a gazdagép busza és a 3. ábra 314 puffertárának belső RAM-ja között. A 312 DMA vezérlő és arbiter nyolcpontos Grey-kódú sorrendvezérlőt használ, amelyet 30 ns-os órajellel ütemezünk, és amely 240 ns-os löketciklusokat szolgáltat. A 312 DMA vezérlő és arbiter tartalmazza a gazdarendszerbusz lekérdezésére, bájt kiigazításra, adatvezérlésre, és tartásra a modulomemória-címzéshez szükséges logikákat, valamint a belső puffertár-csomagoló logikát. Ugratás alatt azt értjük, hogy a cím növelését a memória egyik kezdőcímről a rá következő kezdőcímré ugorva határozzuk meg. Tartás alatt a szavak (címek) számát értjük egy kezdőcímtől kiindulva, amelyeket egyszerre kell kiolvasni. Átugrás alatt azoknak a szavaknak (címeknek) a számát értjük, amelyet nem olvasunk, azaz azoknak a szavaknak a számát, amelyeknél az ugrás meghaladja a tartásértékeket. A 312 DMA vezérlő és arbiter kapcsolatot biztosít a 310 csomagvezérlőhöz és a 311 DMA kezelőhöz a 3. ábrán látható módon. A 310 csomagvezérlő interfésze egy-

szerűen egy vezérlőszó-interfész, és a 4. ábra vezérlőszó-vának vezérlőszó-paraméterei igazodnak a gazdabuszmesterhez és az arbiterhez. Ezek a 3. ábrán láthatóan magukban foglalják az iránybitek, a bájtűzjelzőt, a rendszer felső és alsó címeit stb. Az iránybitek tudatják a 312 DMA vezérlő és arbiterrel, hogy olvasás- vagy írásműveletet kell végrehajtania a gazdarendszer-memóriában. Egy rendszer M/I/O bit tudatja a 312 DMA vezérlő és arbiterrel, hogy az írás vagy az olvasás a gazdagép memóriájában vagy annak valamilyen I/O területén hajtható végre. A felső és alsó rendszercímek együttesen meghatározzák a 312 DMA vezérlő és arbiter buszmesterműveletekre használt kezdőcímét. A rendszercím-számláló 20 biten túl növelhető, így az összes művelet 1 megabájt határon van elrendezve. Az ugrás/tartás és engedélyezésbitek, amelyeket a 310 csomagvezérlő ad át a 312 DMA vezérlő és arbiternek, lehetővé teszik a rendszergazdacím-növelő számára az ugrást, azaz engedélyezés esetén egy modulo- vagy blokkmemória-címzés megtételét. Az ugrás- és tartásértékek a 310 csomagvezérlő által a 312 DMA vezérlő és arbiter részére átadott bájtértékek. A „tartás”-érték bele van építve az „ugrás”-értékbe a korábban kifejtett „tartás+kihagyás=ugrás” általános képletben.

A 311 DMA kezelő indítja a 312 DMA vezérlő- és arbiterrendszer buszmestervezérlő és arbiterműveleteit. Megindítás után a 312 DMA vezérlő és arbiter teljes felügyeletet gyakorol a belső RAM 314 puffertárra, amelyhez megfelelő puffertárcímeket biztosít és írásengedélyezés-jeleket és jóváhagyja vagy fogadja a puffertár-adatokat. A löketszám- és az új csomagszám-paramétereket a 11 DMA kezelő továbbítja a 312 DMA vezérlő és arbiterhez.

A 310 csomagvezérlő általános működését a második DSP processzorból érkező, korábban „léptető” vezérlésként jelölt jel indítja el. A 9. ábrán megfigyelhető a 3. ábra 310 csomagvezérlőjének a működése. A 310 csomagvezérlő a második DSP processzorból érkező léptetőjel hatására kezdi meg működését. 40 lépésben megszólítja a második DSP processzor adatmemóriáját, amelyben a DMA csomaglista található, és kiolvassa a ciklusszámlálót. Ezzel 41 lépésben megkapja az első DMA csomagmutató-kezdőpozíciót a második DSP processzortól, majd 42 lépésben ismét kiolvassa a második DSP processzor adatmemóriáját és a ciklusszámlálót. 43 lépésben a mutatót megnöveljük, és 44 lépésben egy tesztet végzünk el, hogy megállapítsuk, vajon elértük-e a második DSP processzor DMA csomagkérelem-listában a csomaglista végét. Ha a csomaglista végét még nem értük el, a 310 csomagvezérlő a 45 lépésben kiolvassa az iránybitek, és ha mindkét iránybit értéke nulla, jelezve, hogy „várakozóállapot”-csomag érkezett, akkor 46 lépésben négyszer megnöveli a csomagszámláló értékét, majd visszatér a 40 lépésre, egy újabb léptetőjelre várva.

Ha a „várakozóállapot”-bitek nem észleli, úgy a művelet 48 lépésben folytatódik, amelyben a vezérlőregisztereket kiolvassuk, úgy, ahogy azt a csomagmutatónál jeleztük. 49 lépésben a csomagmutató értékét megnöveljük, és ellenőrizzük a magas értékű biteket, hogy

azok értéke nagyobb-e nullánál. Ha a magas értékű bitek értéke nullánál nagyobb, 51 lépésben nem töltjük be a tartás- és ugrásparemetereket. Ha azonban a 11–15 bitek értéke nem nulla, a tartás- és ugrásértékeket betöltjük az 1 és 2 vezérlőszó-regiszterek 11–15 bitjei által tartalmazott bitek címekre, amint az a 4. ábrán részletesen is látható.

53 lépésben a 08–23 rendszercímregiszter-bitek feltöltjük azokkal az adatokkal, amelyeket a mutató megjelöl, amelyet aztán 54 lépésben értékében megnövelünk, és a rendszercímét bájt címként eltároljuk. A rendszercím 55 lépésben 00–07 regiszterbitek feltöltjük bármely, a mutató által jelzett adattal, majd a mutatót 56 lépésben értékében megnöveljük, és a meghatározott második DSP processzor címregiszterét megtöltjük azokkal az adatokkal, amelyeket a mutató kijelöl. A mutatót ezt követően 58 lépésben értékében ismét megnöveljük.

59 lépésben a második DSP processzor címet szócímként eltároljuk, és 60 lépésben végrehajtjuk az átvitelt arról a második DSP processzor címről vagy arra a második DSP processzor címről, amelyet a 3. ábra 311 DMA kezelője inicializálása során kijelöltünk. A művelet folytatódik, és visszatérve a 42 lépésre kiolvassuk a következő második DSP processzor adatmemória-címét. Újból elvégezzük az ellenőrzéseket a 44 lépésben a lista vége jelölésért, és ha azt elértük, 47 lépésben a DMA mutatócímet a csomagmutató-kezdőpozícióval betöltjük úgy, hogy a csomagvezérlő visszahurkolódik a második DSP processzor által felépített szegmentált DMA kérelemlista következő partíciójára.

Az eddig leírtakból szakember számára világos, hogy egy olyan általános számítógéprendszert írtunk le, amely egy DMA alrendszert, valamint egy gazda – DMA és egy DMA – második DSP processzor alrendszer-készletet tartalmaz, amelyek egyedileg illeszkednek a szoros valós idejű konkurens adatfeldolgozási feladatok igényeihez, olyan alkalmazások esetében, amelyeket leggyakrabban multimédia-számítógéprendszerekben futtatnak. Az általános multimédia-számítógéprendszer elemei jól ismertek és kereskedelmi forgalomban kaphatók, beleértve a második DSP processzorokat, 20 DMA vezérlő- és arbitercsipeket és gazdagépeket.

A rendszer és az alrendszerek konfigurálásának, valamint az alrendszerek és az egyes processzorok közötti kommunikációnak és vezérlésnek köszönhető az az egyedülálló előnyök, amelyeket a találmány szerinti rendszer biztosít. A szakterületen jártas olvasó számára ezért nyilvánvaló, hogy a rendszer és az alrendszerek hardver- és szoftverrészén számos olyan módosítás elvezhető, amely részben megváltoztatja a találmány felépítését vagy működését, de a találmány által körülhatárolt és a szabadalmi igénypontokban megfogalmazott oldalmi igényt nem lépi túl. Így például, míg a jelenleg kapható jelfeldolgozóprocesszor-sebességek és memóriakapacitások a célnak teljesen megfelelnek, a közeljövőben lényegesen gyorsabb processzorok és lényegesen nagyobb memóriakapacitások válnak elérhetővé bárki számára, minek következtében az erőforrás-kezelő funkció gyakorlati szempontok alapján előnyösen áttevődik



a digitális processzortól a második DSP processzorhoz, vagy pedig az előzővel ellentétes módon a második DSP processzor erőforrásigény-kezelését egy külső segédprocesszorra bízhatjuk, ugyanabban a rendszerben, amelyben a digitális processzor és a második DSP processzor található, vagy pedig a második DSP processzor olyan sebességű és olyan memóriakapacitású lehet, ami lehetővé teszi, hogy ne csupán a második DSP processzor feladatait lássa el, hanem kezelni tudja saját erőforrásai lefoglaltságát és az aktuálisan futtatott felhasználói alkalmazásokat is a belső processzor helyett, anélkül, hogy a találmány lényegétől és oltalmi körétől eltérnénk. Ugyanígy a DMA-t teljesen kiiktathatjuk és helyettesíthetjük, ha a második DSP processzor átveszi a felhasználóalkalmazás-programok közvetlen végrehajtását, így a 20 DMA vezérlő és arbiterre sem lesz szükség, mivel a második DSP processzor operációs rendszere kezelni tudja és allokálja a második DSP processzor jelfeldolgozó erőforrásait a kezelt feladatlistához éppúgy, ahogy a 20 DMA vezérlő és arbiter és a második DSP processzor a csomaglistákat kezeli a bemutatott előnyös kiviteli alakban. Így például a második DSP processzor csomaglista-kérelmeit saját maga is lekezelheti és feldolgozhatja, mint ahogy a kiindulási erőforrás-allokációs feladatot is elvégezhetné, ha kielégítő sebességű és memóriakapacitású második DSP processzor kapható lenne. Röviden, a találmány itt egyesíti magában a felhasználói alkalmazások gyors végrehajtását, az erőforrás-allokációt, az adatátviteli lista létrehozást, jelfeldolgozó feladatok végrehajtását a felhasználói alkalmazások segítésére, valamint a DMA funkció teljes helyettesítését egyetlen második DSP processzor felhasználásával, azonban lényegesen nagyobb végrehajtási sebességgel és/vagy nagyobb memóriakapacitással, mint a második DSP processzorban szokás.

## SZABADALMI IGÉNYPONTOK

1. Multimédia-számítógéprendszer, amely felhasználóalkalmazás-programok végrehajtásához egy, a felhasználóalkalmazás-programok végrehajtásának támogatása során a rendszerhez digitális jelfeldolgozási kérelmeket továbbító első processzort, továbbá a felhasználóalkalmazás-programok első processzor általi végrehajtásának támogatása során a digitális jelfeldolgozási kérelmeket végrehajtó második processzort, továbbá az utóbbi által fogadott digitális jelfeldolgozási kérelmek függvényében a második processzort a közvetlen memória-hozzáférésű adatátviteli kérelmek particionált várakozósorának a digitális jelfeldolgozási kérelmek támogatása során az adatok második processzor felé vagy processzortól való továbbítása céljából történő létrehozására utasító eszközt tartalmaz, *azzal jellemezve*, hogy az első processzort a csupán együttes jelfeldolgozásierőforrás-igényükkel a második processzor (DSP) rendelkezésre álló jelfeldolgozásierőforrás-kapacitását meg nem haladó digitális jelfeldolgozási kérelmek második processzorhoz (DSP) továbbítására utasító eszköze van, valamint a particionált várakozósor a

második processzor (DSP) által szabályosan visszatérő időközönként behelyezett particiójelöléseket tartalmaz.

2. Az 1. igénypont szerinti multimédia-számítógéprendszer, *azzal jellemezve*, hogy a közvetlen memória-hozzáférésű (DMA) adatátviteli kérelmeket a memória-forrás- és célcímeket, átviteli irányt, átvendő adategységek számát, és működési vezérlőjelöléseket, valamint kérelem vége jelölést meghatározó adatokat tartalmazó mezőkkel ellátó második processzort (DSP) tartalmaz.

3. Az 1. vagy 2. igénypont szerinti multimédia-számítógéprendszer, *azzal jellemezve*, hogy az első processzor és a második processzor (DSP) közötti adatmozgatást vezérlő DMA vezérlő és arbiter (20), valamint azt az első processzorral és a második processzorral (DSP) összekötő adatbuszt és címbuszt tartalmaz.

4. A 2. vagy 3. igénypont szerinti multimédia-számítógéprendszer, *azzal jellemezve*, hogy a particionált várakozósor elérése, valamint az adatmozgásnak a kérelembe épített adatokat tartalmazó mezővel összhangban történő mozgatásának vezérlése révén az első processzor és a második processzor (DSP) között adatátvitelt végrehajtó processzor közti DMA vezérlő és arbiterrel (20) rendelkezik.

5. A 3. vagy 4. igénypont szerinti multimédia-számítógéprendszer, *azzal jellemezve*, hogy egy várakozósor-partíciót létrehozó, és abba a várakozósor másik particiójából származó adatátviteli kérelmeken alapuló adatátvitelt végrehajtó processzor közti DMA vezérlő és arbiter (20) adatátviteli kérelmeit szabályos időközönként elhelyező második processzorral (DSP) rendelkezik.

6. Az 1-5. igénypontok bármelyike szerinti multimédia-számítógéprendszer, *azzal jellemezve*, hogy a felhasználó által kiválasztott minden egyes felhasználóalkalmazás-program támogatásához szükséges második processzor (DSP) jelfeldolgozásierőforrás-igényekhez a felhasználó által kiválasztott alkalmazási program meghatározott időintervallum során jelentkező legnagyobb adategység-átviteli igényét, a legnagyobb igényelt végrehajtási sebességet és a második processzor (DSP) által igényelt legnagyobb memóriamennyiséget magában foglaló hozzáférést megvalósító eszközei vannak.

7. A 4. igénypont szerinti multimédia-számítógéprendszert, *azzal jellemezve*, hogy az adatátviteli kérelmeket a memória-forráscím elérésével, az ott lévő adatoknak a puffertár-memóriába való átmozgatásával, a memóriacélcím elérésével, valamint a puffertárból az abban lévő adatoknak a célcímre való mozgatásával végrehajtó processzor közti DMA vezérlő és arbiter (20) puffertár-memóriát tartalmaz.

8. Eljárás multimédia-számítógéprendszer működésének vezérlésére, amely felhasználóalkalmazás-programokat végrehajtó első processzort és a felhasználóalkalmazás-programoknak az első processzorban való végrehajtásának támogatása során digitális jelfeldolgozási kérelmeket végrehajtó második processzort tartalmaz, *azzal jellemezve*, hogy felhasználó által kiválasztott aktív alkalmazási programok végrehajtásának támogatására meghatározzuk a második processzor (DSP) teljes erőforrás-igényét és összehasonlítjuk a második processzor (DSP) legnagyobb rendelkezésre álló erőforrás-kapaci-



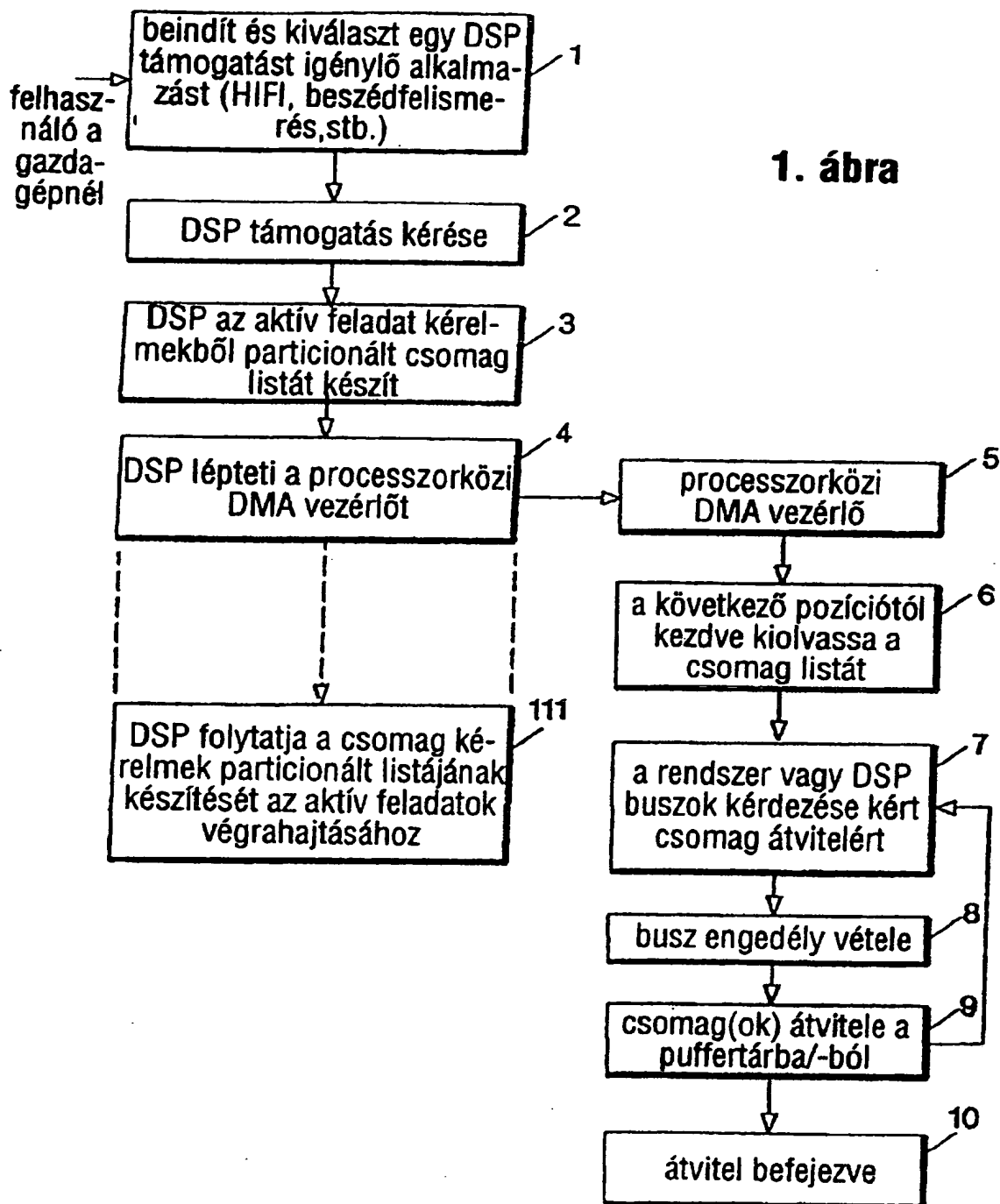
tásával, és ha a felhasználóalkalmazás-programok erőforrásigénye következtében a második processzor (DSP) teljes erőforrásigénye meghaladja a második processzor (DSP) legnagyobb rendelkezésre álló erőforrás-kapacitását, akkor megtöltjük a kiválasztott felhasználóalkalmazás-program első processzor általi végrehajtását, továbbá a második processzorral (DSP) vett digitális jelfeldolgozási kérelmek függvényében közvetlen memóriahozzáférésű (DMA) adatátviteli kérelem particionált várakozósort hozunk létre, amelyben szabályosan ismétlődő időközönként particiójelöléseket helyezünk el.

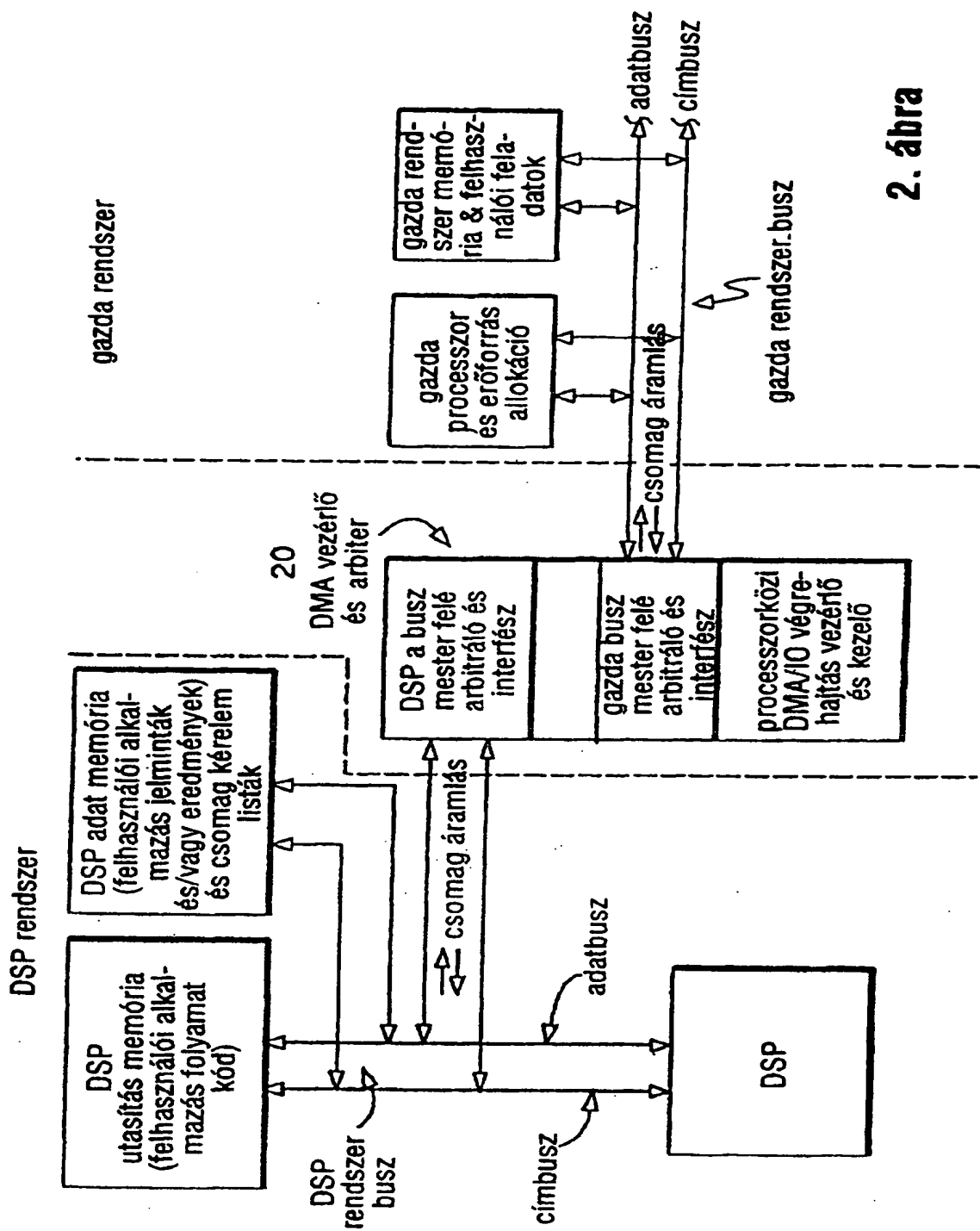
5

9. A 8. igénypont szerinti eljárás, *azzal jellemezve*, hogy az első processzorhoz vagy a második processzorhoz (DSP) továbbítandó adatátvitelikérelem-adatokat átmeneti puffertár-memóriába mozgatjuk, majd abból mozgatjuk át az első processzor vagy a második processzor (DSP) célmemóriacímére.

10

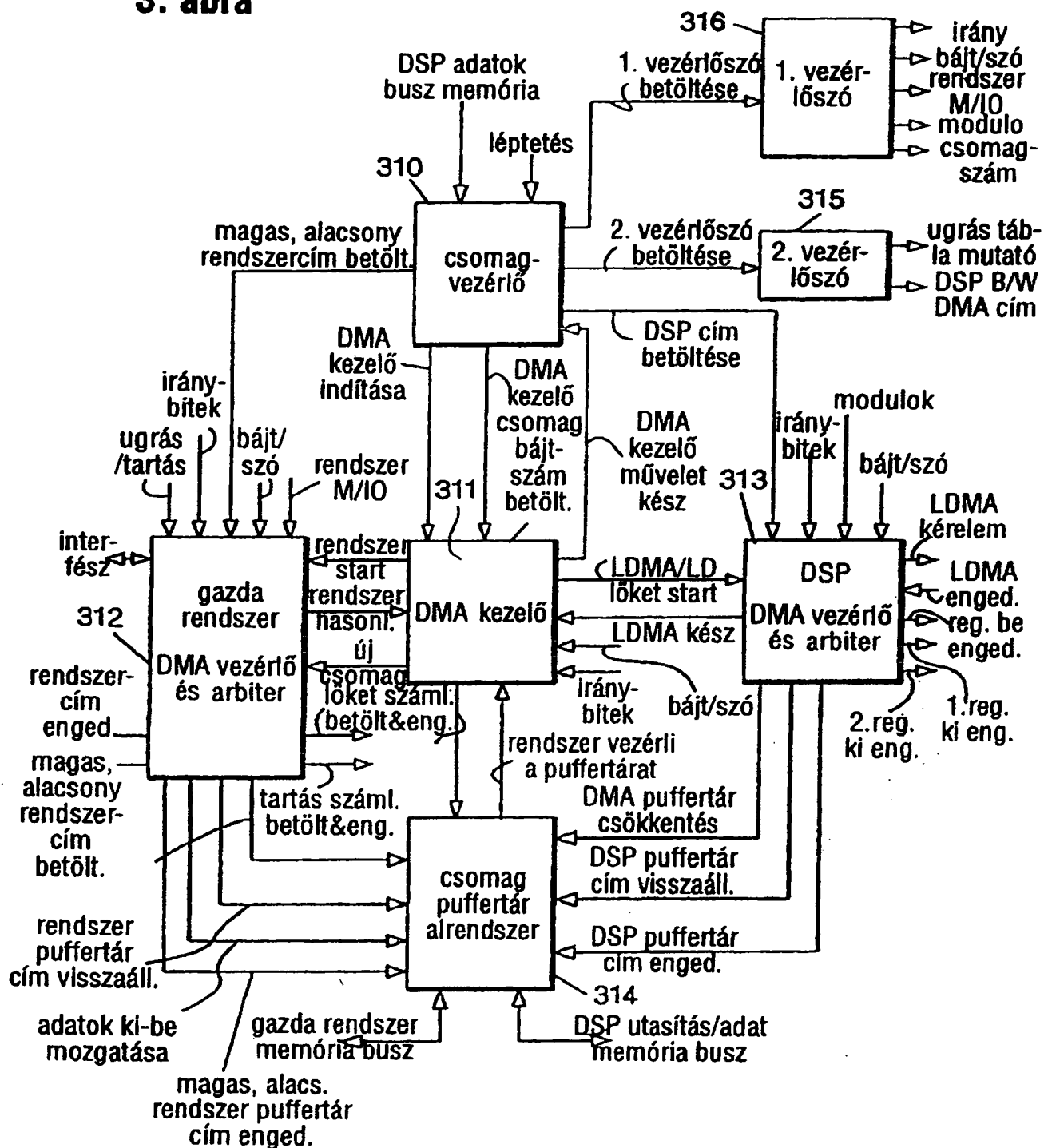
10. A 8. vagy 9. igénypont szerinti eljárás, *azzal jellemezve*, hogy a várakozósor particiójában lévő adatátviteli kérelmeket azon időközön belül hajtjuk végre, miközben a második processzorral (DSP) a várakozósor egy másik particióját hozzuk létre.

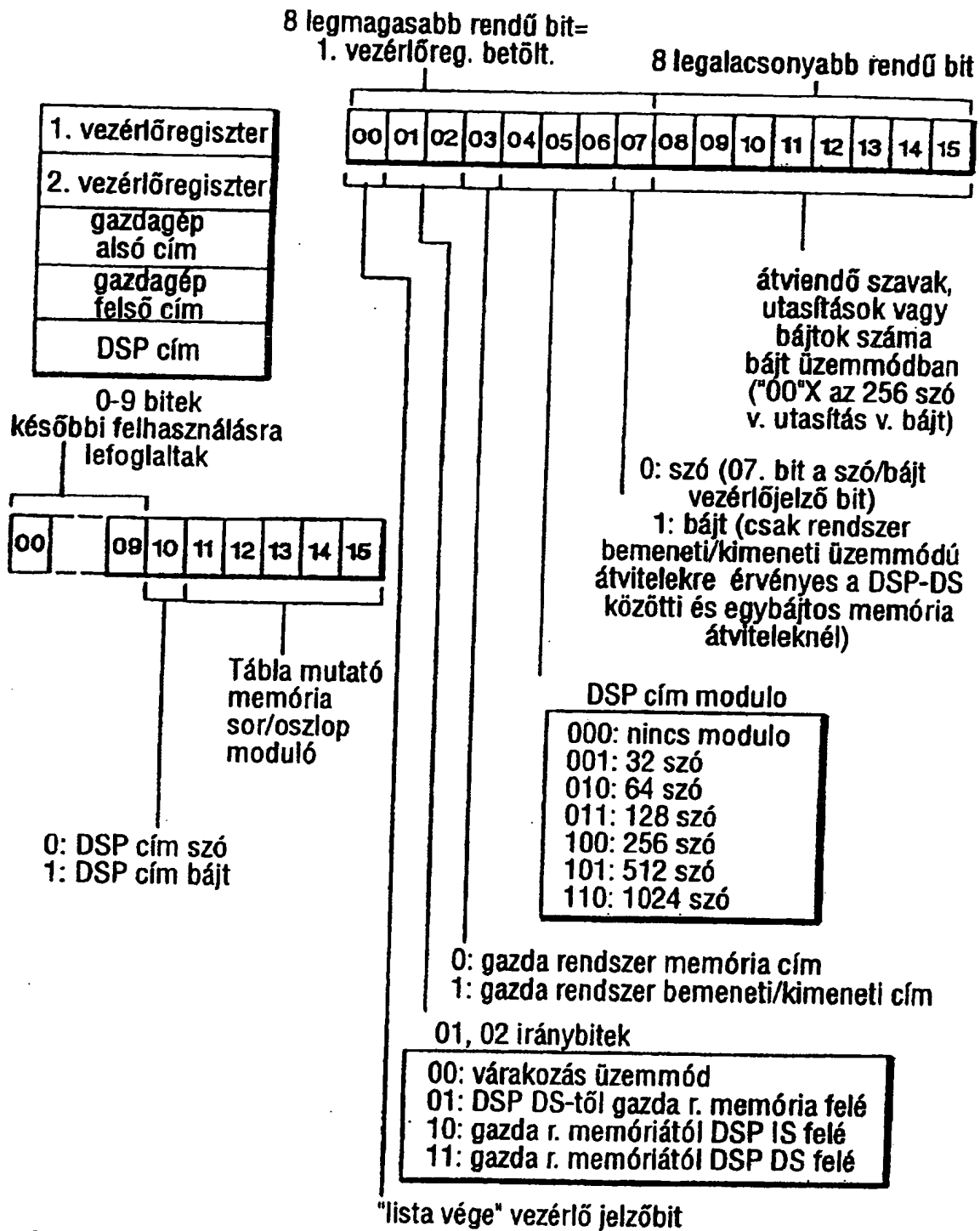




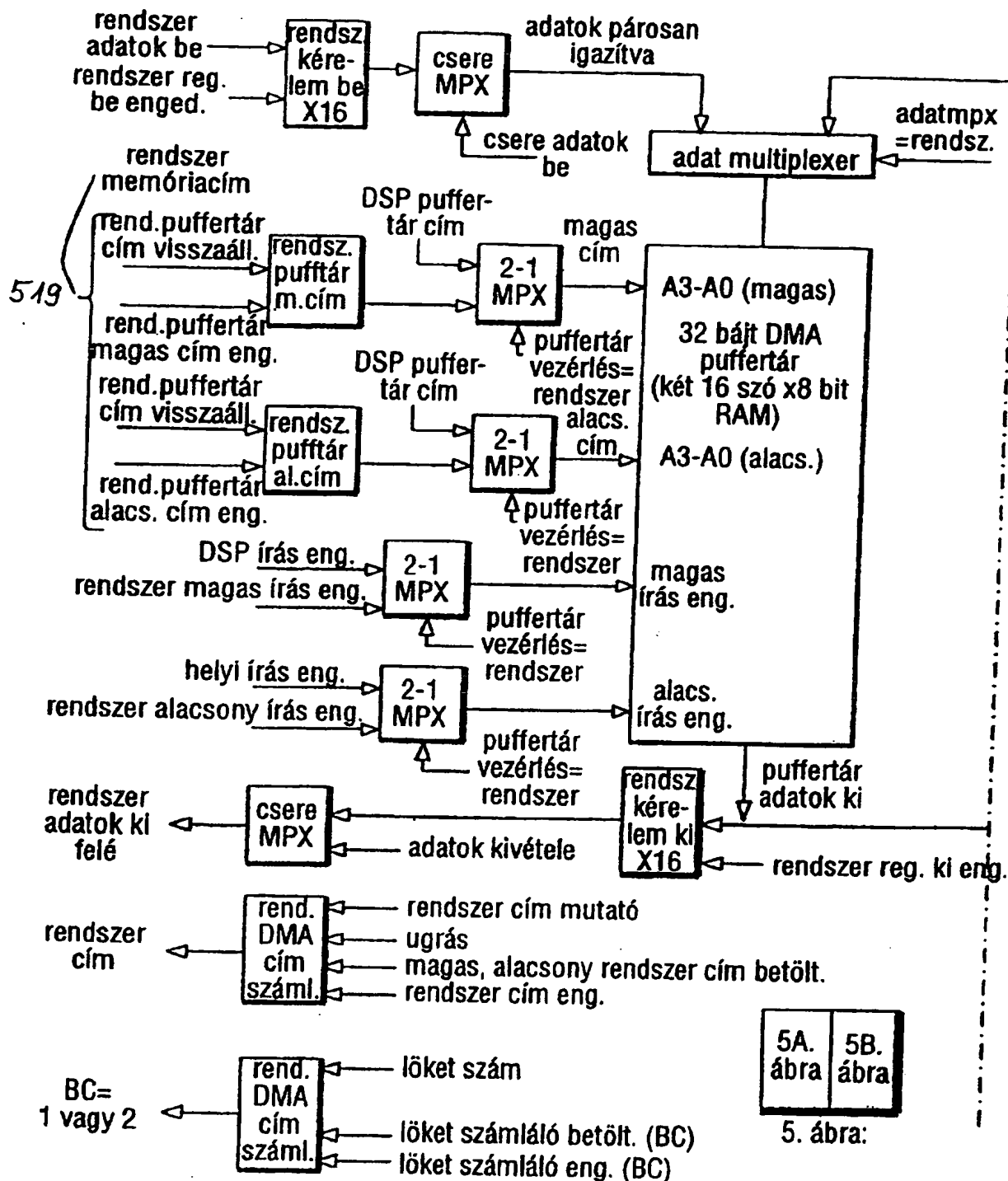
2. ábra

### 3. ábra

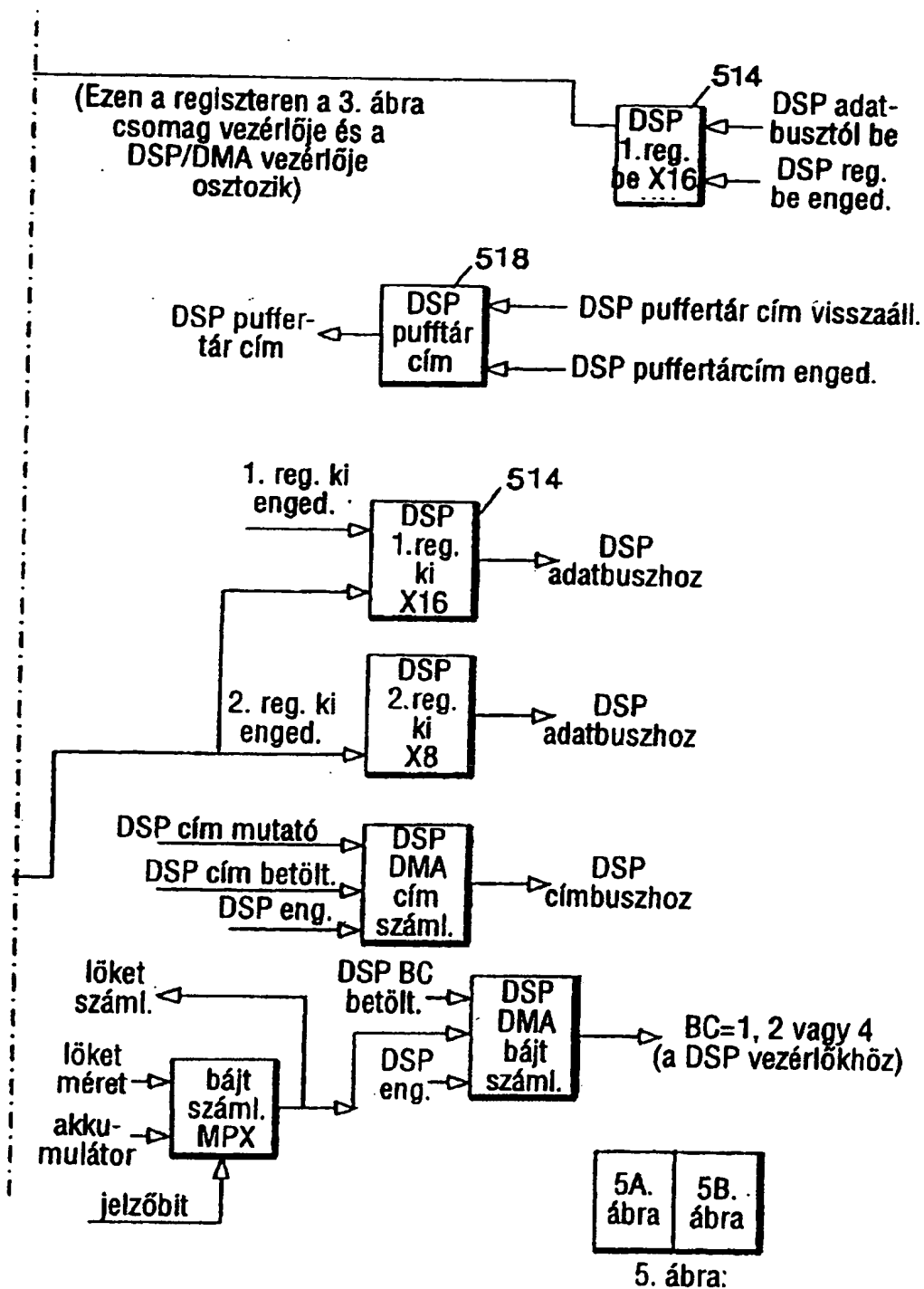




4. ábra

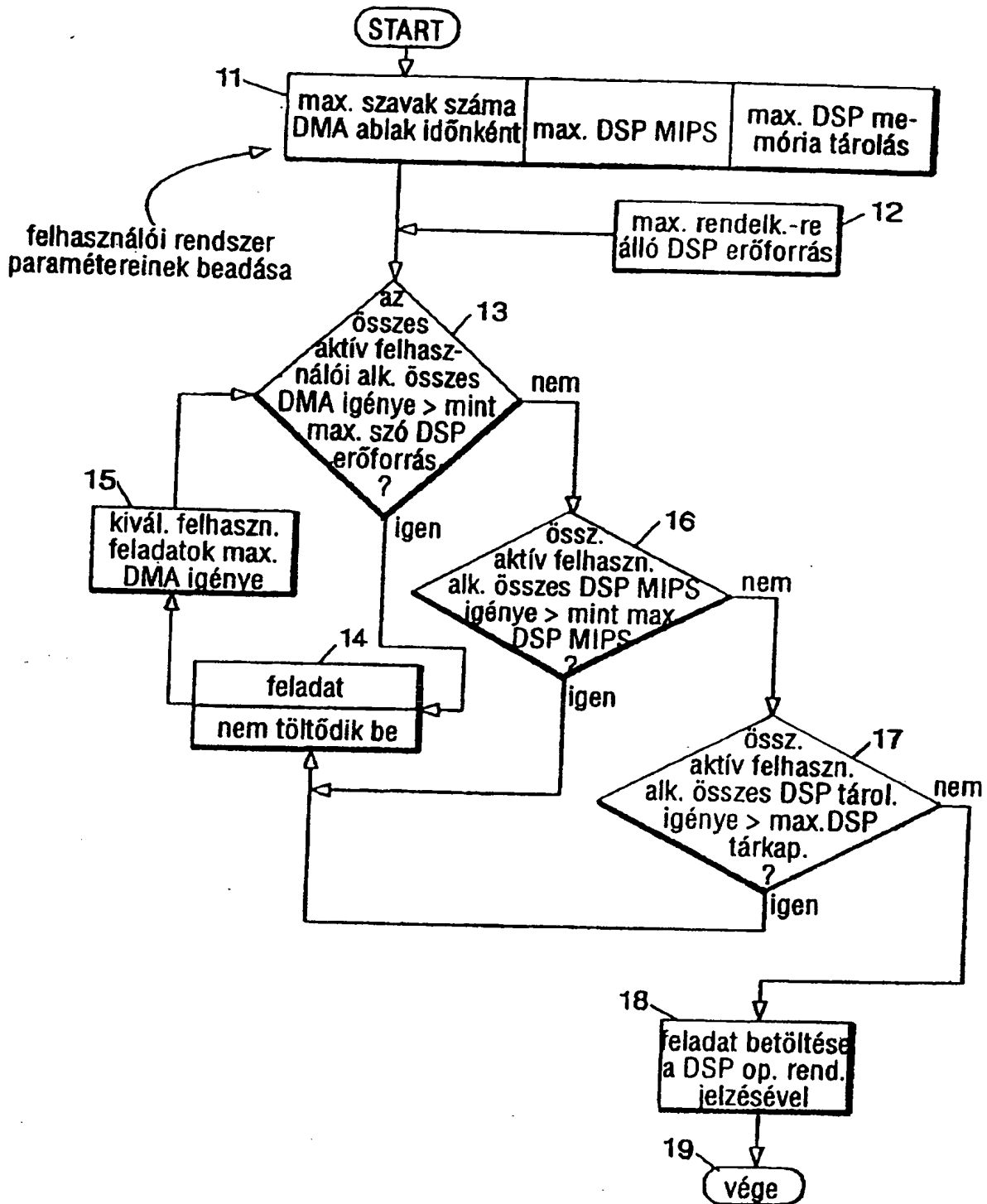


5A. ábra

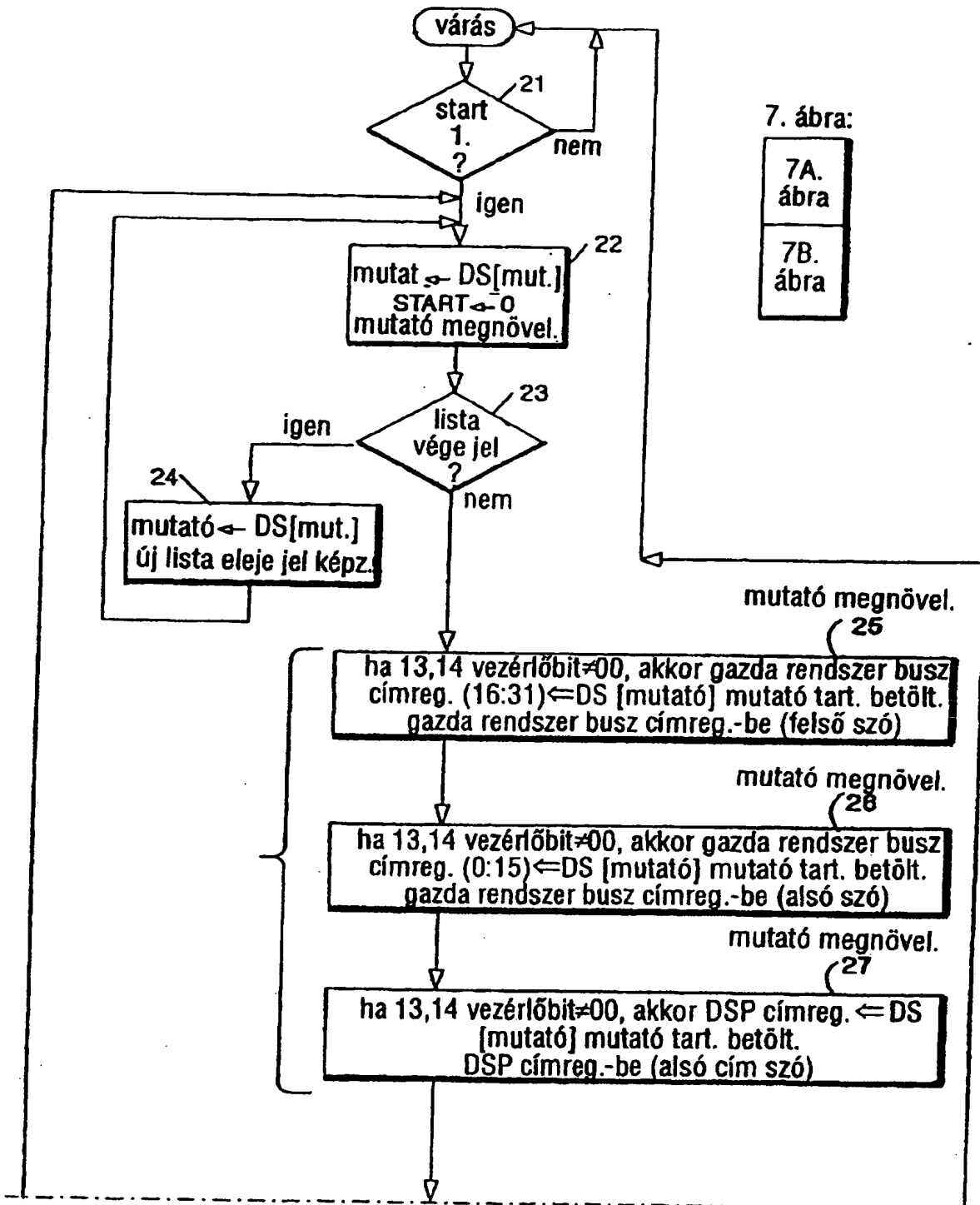


5B. ábra

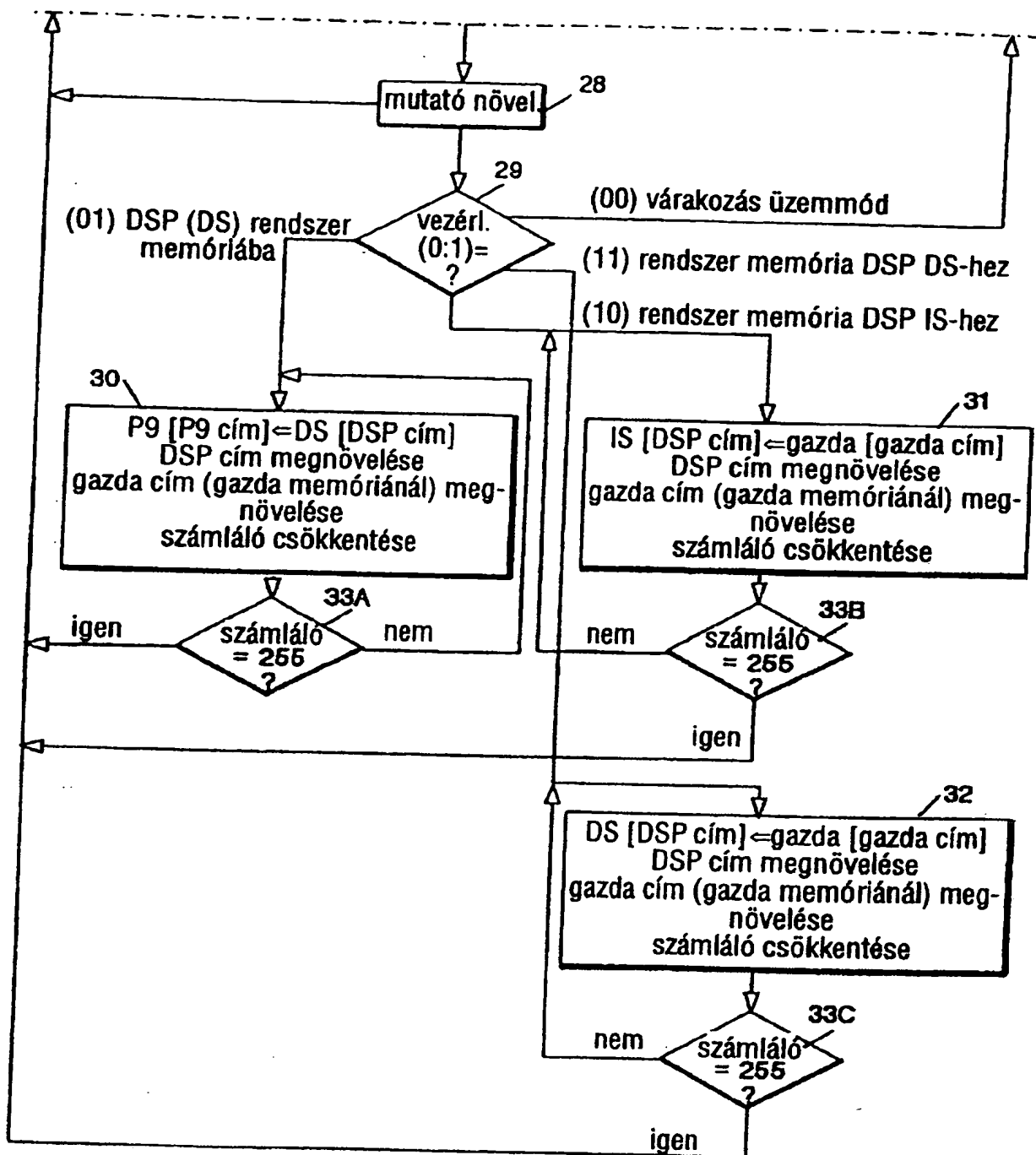




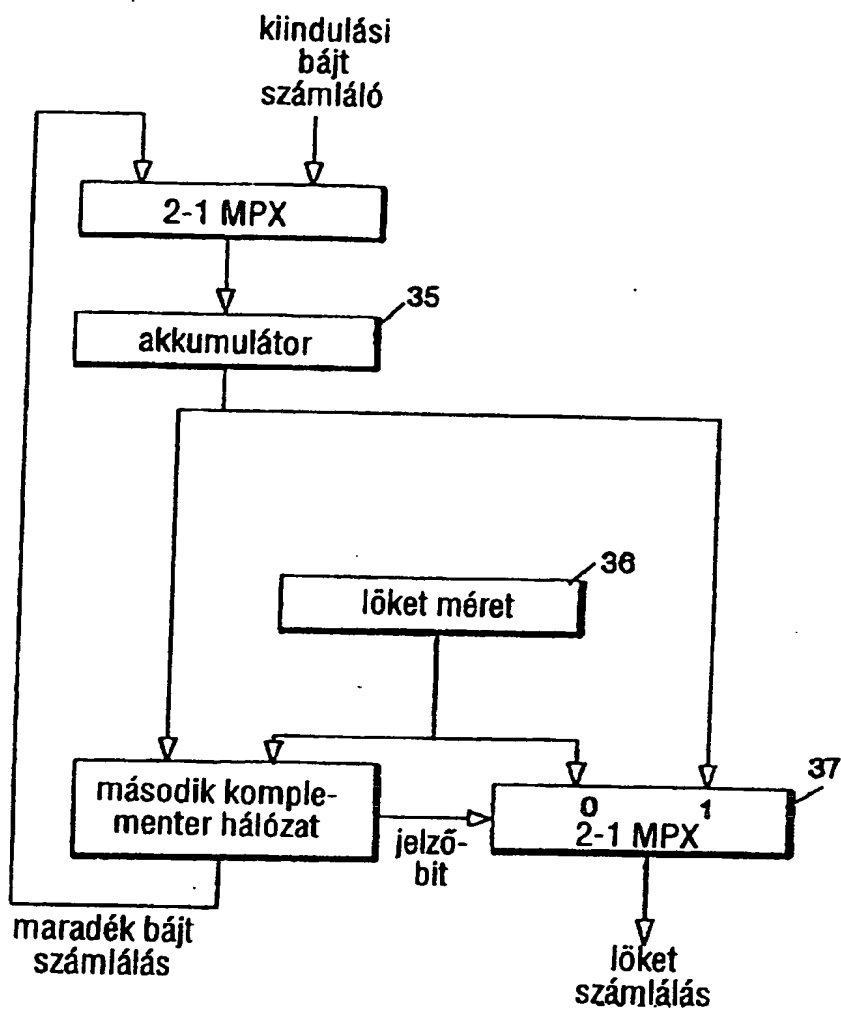
6. ábra



7A. ábra



7B. ábra

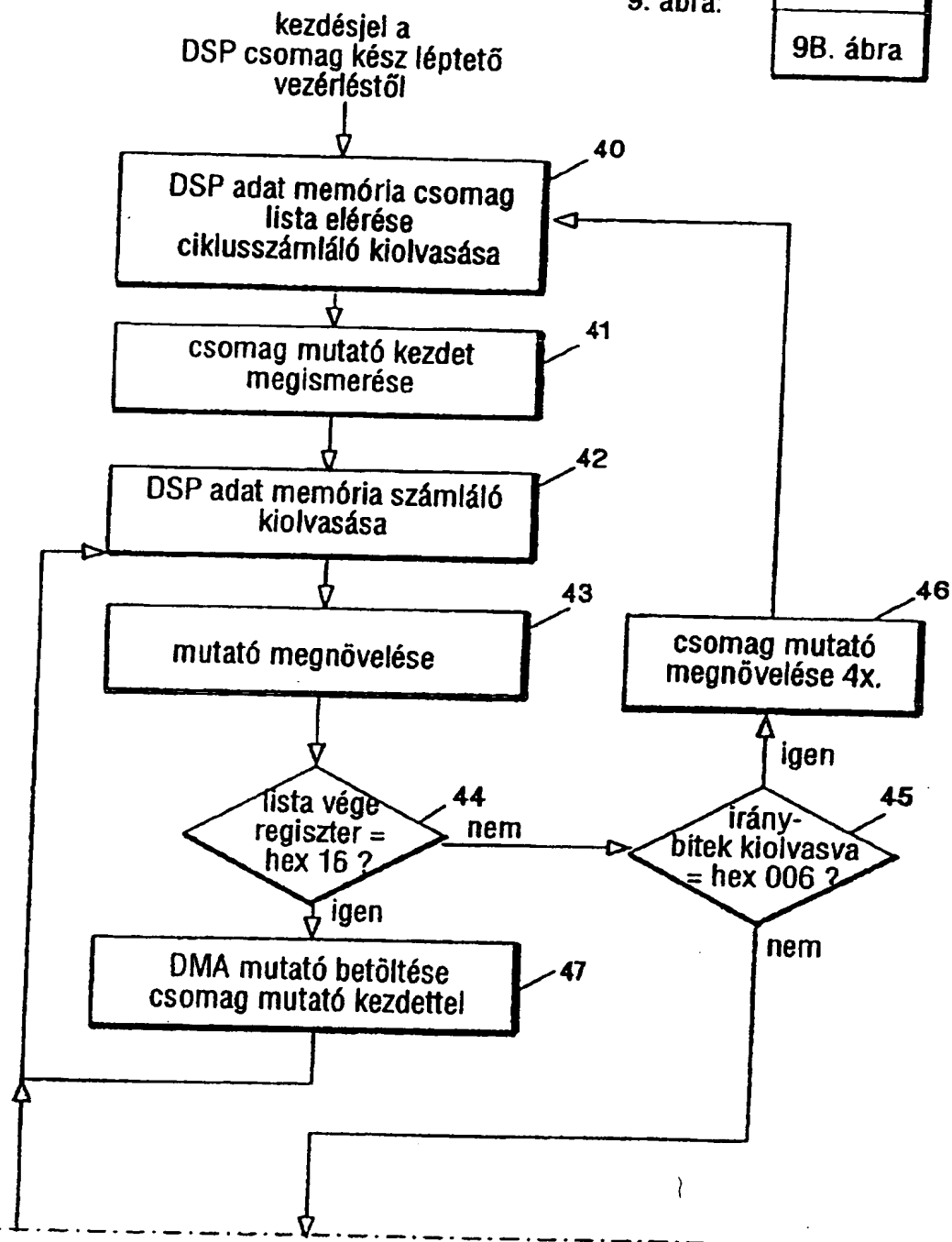


8. ábra

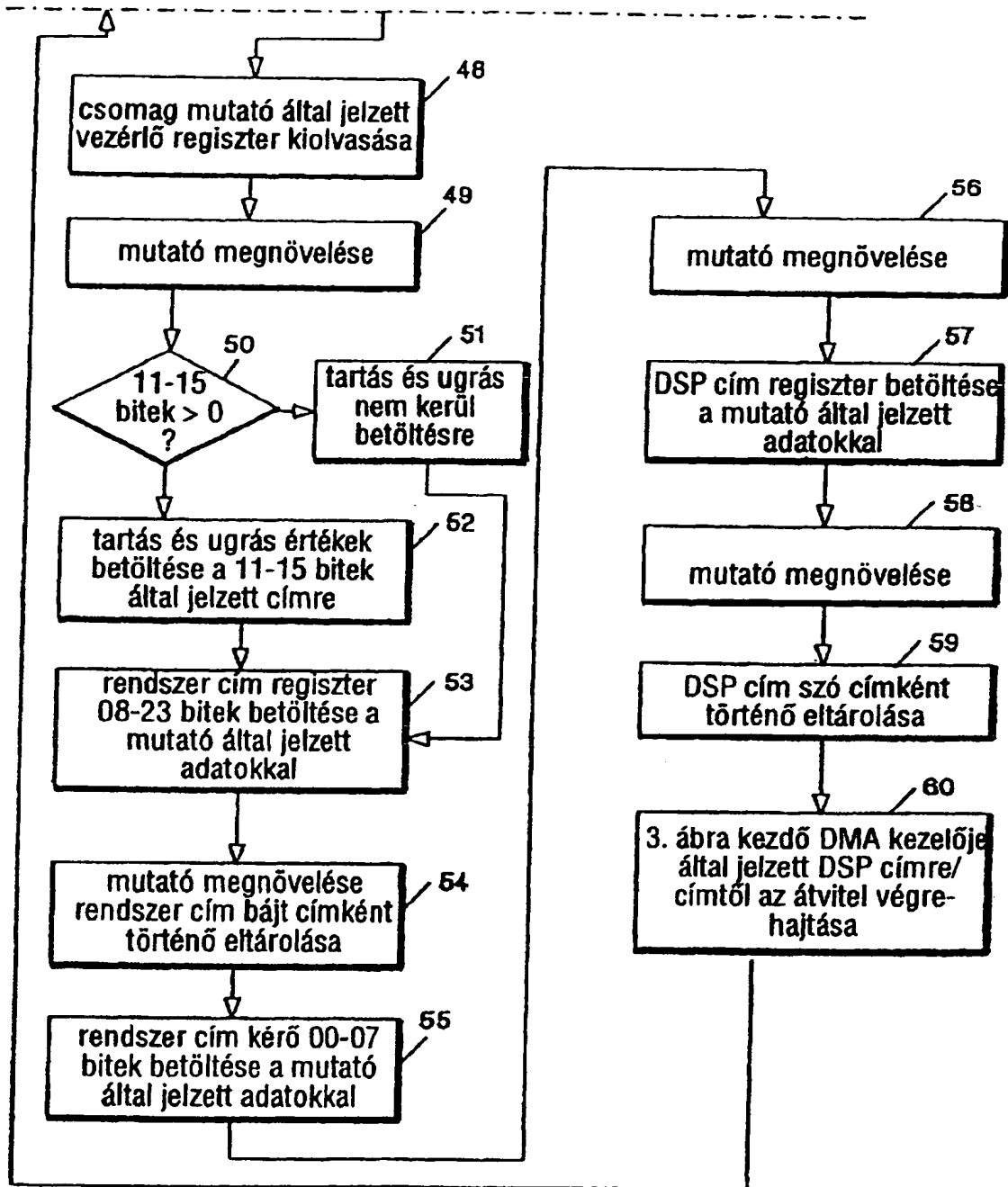
9. ábra:

9A. ábra

9B. ábra



9A. ábra



9B. ábra

